

Network Working Group
Request for Comments: 4006
Category: Standards Track

H. Hakala
L. Mattila
Ericsson
J-P. Koskinen
M. Stura
J. Loughney
Nokia
August 2005

Diameter Credit-Control Application

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document specifies a Diameter application that can be used to implement real-time credit-control for a variety of end user services such as network access, Session Initiation Protocol (SIP) services, messaging services, and download services.

Table of Contents

| | | |
|------|--|----|
| 1. | Introduction..... | 4 |
| 1.1. | Requirements Language..... | 5 |
| 1.2. | Terminology..... | 5 |
| 1.3. | Advertising Application Support..... | 7 |
| 2. | Architecture Models..... | 7 |
| 3. | Credit-Control Messages..... | 9 |
| 3.1. | Credit-Control-Request (CCR) Command..... | 9 |
| 3.2. | Credit-Control-Answer (CCA) Command..... | 11 |
| 4. | Credit-Control Application Overview..... | 11 |
| 4.1. | Service-Specific Rating Input and Interoperability.... | 13 |
| 5. | Session Based Credit-Control..... | 15 |
| 5.1. | General Principles..... | 15 |
| 5.2. | First Interrogation..... | 21 |
| 5.3. | Intermediate Interrogation..... | 27 |
| 5.4. | Final Interrogation..... | 29 |

| | | |
|-------|---|----|
| 5.5. | Server-Initiated Credit Re-Authorization..... | 30 |
| 5.6. | Graceful Service Termination..... | 32 |
| 5.7. | Failure Procedures..... | 38 |
| 6. | One Time Event..... | 41 |
| 6.1. | Service Price Enquiry..... | 42 |
| 6.2. | Balance Check..... | 42 |
| 6.3. | Direct Debiting..... | 43 |
| 6.4. | Refund..... | 44 |
| 6.5. | Failure Procedure..... | 44 |
| 7. | Credit-Control Application State Machine..... | 46 |
| 8. | Credit-Control AVPs..... | 55 |
| 8.1. | CC-Correlation-Id AVP..... | 58 |
| 8.2. | CC-Request-Number AVP..... | 58 |
| 8.3. | CC-Request-Type AVP..... | 58 |
| 8.4. | CC-Session-Failover AVP..... | 59 |
| 8.5. | CC-Sub-Session-Id AVP..... | 59 |
| 8.6. | Check-Balance-Result AVP..... | 60 |
| 8.7. | Cost-Information AVP..... | 60 |
| 8.8. | Unit-Value AVP..... | 61 |
| 8.9. | Exponent AVP..... | 61 |
| 8.10. | Value-Digits AVP..... | 61 |
| 8.11. | Currency-Code AVP..... | 62 |
| 8.12. | Cost-Unit AVP..... | 62 |
| 8.13. | Credit-Control AVP..... | 62 |
| 8.14. | Credit-Control-Failure-Handling AVP..... | 62 |
| 8.15. | Direct-Debiting-Failure-Handling AVP..... | 63 |
| 8.16. | Multiple-Services-Credit-Control AVP..... | 64 |
| 8.17. | Granted-Service-Unit AVP..... | 65 |
| 8.18. | Requested-Service-Unit AVP..... | 66 |
| 8.19. | Used-Service-Unit AVP..... | 66 |
| 8.20. | Tariff-Time-Change AVP..... | 67 |
| 8.21. | CC-Time AVP..... | 67 |
| 8.22. | CC-Money AVP..... | 67 |
| 8.23. | CC-Total-Octets AVP..... | 68 |
| 8.24. | CC-Input-Octets AVP..... | 68 |
| 8.25. | CC-Output-Octets AVP..... | 68 |
| 8.26. | CC-Service-Specific-Units AVP..... | 68 |
| 8.27. | Tariff-Change-Usage AVP..... | 68 |
| 8.28. | Service-Identifier AVP..... | 69 |
| 8.29. | Rating-Group AVP..... | 69 |
| 8.30. | G-S-U-Pool-Reference AVP..... | 69 |
| 8.31. | G-S-U-Pool-Identifier AVP..... | 70 |
| 8.32. | CC-Unit-Type AVP..... | 70 |
| 8.33. | Validity-Time AVP..... | 70 |
| 8.34. | Final-Unit-Indication AVP..... | 71 |
| 8.35. | Final-Unit-Action AVP..... | 72 |
| 8.36. | Restriction-Filter-Rule AVP..... | 72 |
| 8.37. | Redirect-Server AVP..... | 73 |

| | | |
|--------|--|----|
| 8.38. | Redirect-Address-Type AVP..... | 73 |
| 8.39. | Redirect-Server-Address AVP..... | 74 |
| 8.40. | Multiple-Services-Indicator AVP..... | 74 |
| 8.41. | Requested-Action AVP..... | 74 |
| 8.42. | Service-Context-Id AVP..... | 75 |
| 8.43. | Service-Parameter-Info AVP..... | 76 |
| 8.44. | Service-Parameter-Type AVP..... | 76 |
| 8.45. | Service-Parameter-Value AVP..... | 77 |
| 8.46. | Subscription-Id AVP..... | 77 |
| 8.47. | Subscription-Id-Type AVP..... | 77 |
| 8.48. | Subscription-Id-Data AVP..... | 78 |
| 8.49. | User-Equipment-Info AVP..... | 78 |
| 8.50. | User-Equipment-Info-Type AVP..... | 78 |
| 8.50. | User-Equipment-Info-Value AVP..... | 79 |
| 9. | Result Code AVP Values..... | 79 |
| 9.1. | Transient Failures..... | 79 |
| 9.2. | Permanent Failures..... | 80 |
| 10. | AVP Occurrence Table..... | 80 |
| 10.1. | Credit-Control AVP Table..... | 81 |
| 10.2. | Re-Auth-Request/Answer AVP Table..... | 82 |
| 11. | RADIUS/Diameter Credit-Control Interworking Model..... | 82 |
| 12. | IANA Considerations..... | 85 |
| 12.1. | Application Identifier..... | 86 |
| 12.2. | Command Codes..... | 86 |
| 12.3. | AVP Codes..... | 86 |
| 12.4. | Result-Code AVP Values..... | 86 |
| 12.5. | CC-Request-Type AVP..... | 86 |
| 12.6. | CC-Session-Failover AVP..... | 86 |
| 12.7. | CC-Unit-Type AVP..... | 87 |
| 12.8. | Check-Balance-Result AVP..... | 87 |
| 12.9. | Credit-Control AVP..... | 87 |
| 12.10. | Credit-Control-Failure-Handling AVP..... | 87 |
| 12.11. | Direct-Debiting-Failure-Handling AVP..... | 87 |
| 12.12. | Final-Unit-Action AVP..... | 87 |
| 12.13. | Multiple-Services-Indicator AVP..... | 87 |
| 12.14. | Redirect-Address-Type AVP..... | 88 |
| 12.15. | Requested-Action AVP..... | 88 |
| 12.16. | Subscription-Id-Type AVP..... | 88 |
| 12.17. | Tariff-Change-Usage AVP..... | 88 |
| 12.18. | User-Equipment-Info-Type AVP..... | 88 |
| 13. | Credit-Control Application Related Parameters..... | 88 |
| 14. | Security Considerations..... | 89 |
| 14.1. | Direct Connection with Redirects..... | 90 |
| 15. | References..... | 91 |
| 15.1. | Normative References..... | 91 |
| 15.2. | Informative References..... | 92 |
| 16. | Acknowledgements..... | 93 |
| | Appendix A Credit-Control Sequences..... | 94 |

| | | |
|-------------------------------|----------------|-----|
| A.1. | Flow I..... | 94 |
| A.2. | Flow II..... | 96 |
| A.3. | Flow III..... | 98 |
| A.4. | Flow IV..... | 99 |
| A.5. | Flow V..... | 100 |
| A.6. | Flow VI..... | 102 |
| A.7. | Flow VII..... | 103 |
| A.8. | Flow VIII..... | 105 |
| A.9. | Flow IX..... | 107 |
| Authors' Addresses..... | | 112 |
| Full Copyright Statement..... | | 114 |

1. Introduction

This document specifies a Diameter application that can be used to implement real-time credit-control for a variety of end user services such as network access, Session Initiation Protocol (SIP) services, messaging services, and download services. It provides a general solution to real-time cost and credit-control.

The prepaid model has been shown to be very successful, for instance, in GSM networks, where network operators offering prepaid services have experienced a substantial growth of their customer base and revenues. Prepaid services are now cropping up in many other wireless and wire line based networks.

In next generation wireless networks, additional functionality is required beyond that specified in the Diameter base protocol. For example, the 3GPP Charging and Billing requirements [3GPPCHARG] state that an application must be able to rate service information in real-time. In addition, it is necessary to check that the end user's account provides coverage for the requested service prior to initiation of that service. When an account is exhausted or expired, the user must be denied the ability to compile additional chargeable events.

A mechanism has to be provided to allow the user to be informed of the charges to be levied for a requested service. In addition, there are services such as gaming and advertising that may credit as well as debit a user account.

The other Diameter applications provide service specific authorization, and they do not provide credit authorization for prepaid users. The credit authorization shall be generic and applicable to all the service environments required to support prepaid services.

To fulfill these requirements, it is necessary to facilitate credit-control communication between the network element providing the service (e.g., Network Access Server, SIP Proxy, and Application Server) and a credit-control server.

The scope of this specification is the credit authorization. Service specific authorization and authentication is out of the scope.

1.1. Requirements Language

In this document, the key words "MAY", "MUST", "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [KEYWORDS].

1.2. Terminology

AAA

Authentication, Authorization, and Accounting

AA answer

AA answer generically refers to a service specific authorization and authentication answer. AA answer commands are defined in service specific authorization applications, e.g., [NASREQ] and [DIAMMIP].

AA request

AA request generically refers to a service specific authorization and authentication request. AA request commands are defined in service specific authorization applications e.g., [NASREQ] and [DIAMMIP].

Credit-control

Credit-control is a mechanism that directly interacts in real-time with an account and controls or monitors the charges related to the service usage. Credit-control is a process of checking whether credit is available, credit-reservation, deduction of credit from the end user account when service is completed and refunding of reserved credit that is not used.

Diameter Credit-control Server

A Diameter credit-control server acts as a prepaid server, performing real-time rating and credit-control. It is located in the home domain and is accessed by service elements or Diameter AAA servers in

real-time for purpose of price determination and credit-control before the service event is delivered to the end-user. It may also interact with business support systems.

Diameter Credit-control Client

A Diameter credit-control client is an entity that interacts with a credit-control server. It monitors the usage of the granted quota according to instructions returned by credit-control server.

Interrogation

The Diameter credit-control client uses interrogation to initiate a session based credit-control process. During the credit-control process, it is used to report the used quota and request a new one. An interrogation maps to a request/answer transaction.

One-time event

Basically, a request/answer transaction of type event.

Rating

The act of determining the cost of the service event.

Service

A type of task performed by a service element for an end user.

Service Element

A network element that provides a service to the end users. The Service Element may include the Diameter credit-control client, or another entity (e.g., RADIUS AAA server) that can act as a Credit-control client on behalf of the Service Element. In the latter case, the interface between the Service Element and the Diameter credit-control client is outside the scope of this specification. Examples of the Service Elements include Network Access Server (NAS), SIP Proxy, and Application Servers such as messaging server, content server, and gaming server.

Service Event

An event relating to a service provided to the end user.

Session based credit-control

A credit-control process that makes use of several interrogations: the first, a possible intermediate, and the final. The first interrogation is used to reserve money from the user's account and to initiate the process. The intermediate interrogations may be needed to request new quota while the service is being rendered. The final interrogation is used to exit the process. The credit-control server is required to maintain session state for session-based credit-control.

1.3. Advertising Application Support

Diameter nodes conforming to this specification MUST advertise support by including the value of 4 in the Auth-Application-Id of the Capabilities-Exchange-Request and Capabilities-Exchange-Answer command [DIAMBASE].

2. Architecture Models

The current accounting models specified in the Radius Accounting [RFC2866] and Diameter base [DIAMBASE] are not sufficient for real-time credit-control, where credit-worthiness is to be determined prior to service initiation. Also, the existing Diameter authorization applications, [NASREQ] and [DIAMMIP], only provide service authorization, but do not provide credit authorization for prepaid users. In order to support real-time credit-control, a new type of server is needed in the AAA infrastructure: Diameter credit-control server. The Diameter credit-control server is the entity responsible for credit authorization for prepaid subscribers.

A service element may authenticate and authorize the end user with the AAA server by using AAA protocols; e.g., RADIUS or a Diameter base protocol with a possible Diameter application.

Accounting protocols such as RADIUS accounting and the Diameter base accounting protocol can be used to provide accounting data to the accounting server after service is initiated, and to provide possible interim reports until service completion. However, for real-time credit-control, these authorization and accounting models are not sufficient.

When real-time credit-control is required, the credit-control client contacts the credit-control server with information about a possible service event. The credit-control process is performed to determine potential charges and to verify whether the end user's account balance is sufficient to cover the cost of the service being rendered.

Figure 1 illustrates the typical credit-control architecture, which consists of a Service Element with an embedded Diameter credit-control client, a Diameter credit-control server, and an AAA server. A Business Support System is usually deployed; it includes at least the billing functionality. The credit-control server and AAA server in this architecture model are logical entities. The real configuration can combine them into a single host. The credit-control protocol is the Diameter base protocol with the Diameter credit-control application.

When an end user requests services such as SIP or messaging, the request is typically forwarded to a service element (e.g., SIP Proxy) in the user's home domain. In some cases it might be possible that the service element in the visited domain can offer services to the end user; however, a commercial agreement must exist between the visited domain and the home domain. Network access is an example of a service offered in the visited domain where the NAS, through an AAA infrastructure, authenticates and authorizes the user with the user's home network.

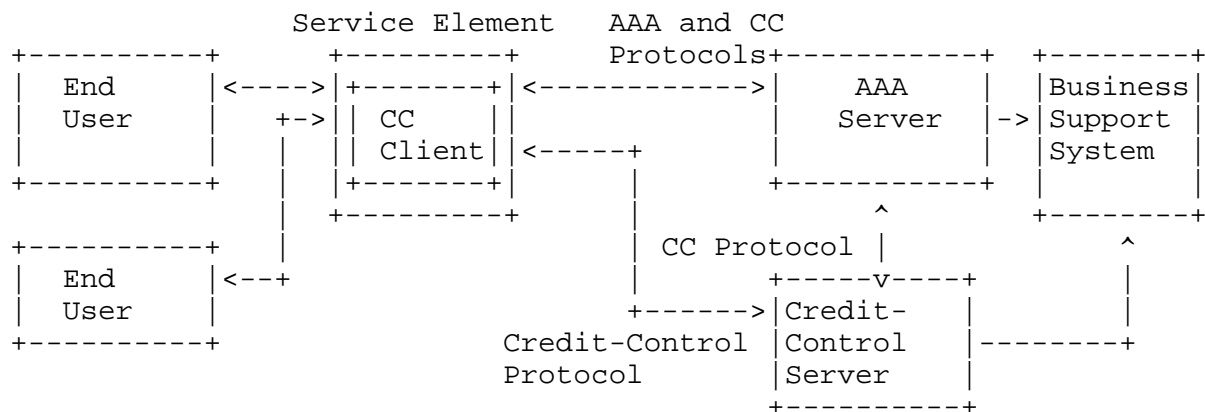


Figure 1: Typical credit-control architecture

There can be multiple credit-control servers in the system for redundancy and load balancing. The system can also contain separate rating server(s), and accounts can be located in a centralized database. To ensure that the end user's account is not debited or credited multiple times for the same service event, only one place in the credit-control system should perform duplicate detection. System internal interfaces can exist to relay messages between servers and an account manager. However, the detailed architecture of the credit-control system and its interfaces are implementation specific and are out of scope of this specification.

Protocol transparent Diameter relays can exist between the credit-control client and credit-control server. Also, Diameter Redirect agents that refer credit-control clients to credit-control servers and allow them to communicate directly can exist. These agents transparently support the Diameter credit-control application. The different roles of Diameter Agents are defined in Diameter base [DIAMBASE], section 2.8.

If Diameter credit-control proxies exist between the credit-control client and the credit-control server, they MUST advertise the Diameter credit-control application support.

3. Credit-Control Messages

This section defines new Diameter message Command-Code values that MUST be supported by all Diameter implementations that conform to this specification. The Command Codes are as follows:

| Command-Name | Abbrev. | Code | Reference |
|------------------------|---------|------|-----------|
| ----- | | | |
| Credit-Control-Request | CCR | 272 | 3.1 |
| Credit-Control-Answer | CCA | 272 | 3.2 |

Diameter Base [DIAMBASE] defines in the section 3.2 the Command Code ABNF specification. These formats are observed in Credit-Control messages.

3.1. Credit-Control-Request (CCR) Command

The Credit-Control-Request message (CCR) is indicated by the command-code field being set to 272 and the 'R' bit being set in the Command Flags field. It is used between the Diameter credit-control client and the credit-control server to request credit authorization for a given service.

The Auth-Application-Id MUST be set to the value 4, indicating the Diameter credit-control application.

Message Format

```
<Credit-Control-Request> ::= < Diameter Header: 272, REQ, PXY >
    < Session-Id >
    { Origin-Host }
    { Origin-Realm }
    { Destination-Realm }
    { Auth-Application-Id }
    { Service-Context-Id }
    { CC-Request-Type }
    { CC-Request-Number }
    [ Destination-Host ]
    [ User-Name ]
    [ CC-Sub-Session-Id ]
    [ Acct-Multi-Session-Id ]
    [ Origin-State-Id ]
    [ Event-Timestamp ]
    *[ Subscription-Id ]
    [ Service-Identifier ]
    [ Termination-Cause ]
    [ Requested-Service-Unit ]
    [ Requested-Action ]
    *[ Used-Service-Unit ]
    [ Multiple-Services-Indicator ]
    *[ Multiple-Services-Credit-Control ]
    *[ Service-Parameter-Info ]
    [ CC-Correlation-Id ]
    [ User-Equipment-Info ]
    *[ Proxy-Info ]
    *[ Route-Record ]
    *[ AVP ]
```

3.2. Credit-Control-Answer (CCA) Command

The Credit-Control-Answer message (CCA) is indicated by the command-code field being set to 272 and the 'R' bit being cleared in the Command Flags field. It is used between the credit-control server and the Diameter credit-control client to acknowledge a Credit-Control-Request command.

Message Format

```
<Credit-Control-Answer> ::= < Diameter Header: 272, PXY >
    < Session-Id >
    { Result-Code }
    { Origin-Host }
    { Origin-Realm }
    { Auth-Application-Id }
    { CC-Request-Type }
    { CC-Request-Number }
    [ User-Name ]
    [ CC-Session-Failover ]
    [ CC-Sub-Session-Id ]
    [ Acct-Multi-Session-Id ]
    [ Origin-State-Id ]
    [ Event-Timestamp ]
    [ Granted-Service-Unit ]
    *[ Multiple-Services-Credit-Control ]
    [ Cost-Information ]
    [ Final-Unit-Indication ]
    [ Check-Balance-Result ]
    [ Credit-Control-Failure-Handling ]
    [ Direct-Debiting-Failure-Handling ]
    [ Validity-Time ]
    *[ Redirect-Host ]
    [ Redirect-Host-Usage ]
    [ Redirect-Max-Cache-Time ]
    *[ Proxy-Info ]
    *[ Route-Record ]
    *[ Failed-AVP ]
    *[ AVP ]
```

4. Credit-Control Application Overview

The credit authorization process takes place before and during service delivery to the end user and generally requires the user's authentication and authorization before any request is sent to the credit-control server. The credit-control application defined in this specification supports two different credit authorization models: credit authorization with money reservation and credit

authorization with direct debiting. In both models, the credit-control client requests credit authorization from the credit-control server prior to allowing any service to be delivered to the end user.

In the first model, the credit-control server rates the request, reserves a suitable amount of money from the user's account, and returns the corresponding amount of credit resources. Note that credit resources may not imply actual monetary credit; credit resources may be granted to the credit control client in the form of units (e.g., data volume or time) to be metered.

Upon receipt of a successful credit authorization answer with a certain amount of credit resources, the credit-control client allows service delivery to the end user and starts monitoring the usage of the granted resources. When the credit resources granted to the user have been consumed or the service has been successfully delivered or terminated, the credit-control client reports back to the server the used amount. The credit-control server deducts the used amount from the end user's account; it may perform rating and make a new credit reservation if the service delivery is continuing. This process is accomplished with session based credit-control that includes the first interrogation, possible intermediate interrogations, and the final interrogation. For session based credit-control, both the credit control client and the credit-control server are required to maintain credit-control session state. Session based credit-control is described in more detail, with more variations, in section 5.

In contrast, credit authorization with direct debiting is a single transaction process wherein the credit-control server directly deducts a suitable amount of money from the user's account as soon as the credit authorization request is received. Upon receipt of a successful credit authorization answer, the credit-control client allows service delivery to the end user. This process is accomplished with the one-time event. Session state is not maintained.

In a multi-service environment, an end user can issue an additional service request (e.g., data service) during an ongoing service (e.g., voice call) toward the same account. Alternatively, during an active multimedia session, an additional media type is added to the session, causing a new simultaneous request toward same account. Consequently, this needs to be considered when credit resources are granted to the services.

The credit-control application also supports operations such as service price enquiry, user's balance check, and refund of credit on the user's account. These operations are accomplished with the one-time event. Session state is not maintained.

A flexible credit-control application specific failure handling is defined in which the home service provider can model the credit-control client behavior according to its own credit risk management policy.

The Credit-Control-Failure-Handling AVP and the Direct-Debiting-Failure-Handling AVP are defined to determine what is done if the sending of credit-control messages to the credit-control server has been temporarily prevented. The usage of the Credit-Control-Failure-Handling AVP and the Direct-Debiting-Failure-Handling AVP allows flexibility, as failure handling for the credit-control session and one time event direct debiting may be different.

4.1. Service-Specific Rating Input and Interoperability

The Diameter credit-control application defines the framework for credit-control; it provides generic credit-control mechanisms supporting multiple service applications. The credit-control application, therefore, does not define AVPs that could be used as input in the rating process. Listing the possible services that could use this Diameter application is out of scope for this generic mechanism.

It is reasonable to expect that a service level agreement will exist between providers of the credit-control client and the credit-control server covering the charging, services offered, roaming agreements, agreed rating input (i.e., AVPs), and so on.

Therefore, it is assumed that a Diameter credit-control server will provide service only for Diameter credit-control clients that have agreed beforehand as to the content of credit-control messages. Naturally, it is possible that any arbitrary Diameter credit-control client can interchange credit-control messages with any Diameter credit-control server, but with a higher likelihood that unsupported services/AVPs could be present in the credit-control message, causing the server to reject the request with an appropriate result-code.

4.1.1. Specifying Rating Input AVPs

There are two ways to provide rating input to the credit-control server: either by using AVPs or by including them in the Service-Parameter-Info AVP. The general principles for sending rating parameters are as follows:

1a. The service SHOULD re-use existing AVPs if it can use AVPs defined in existing Diameter applications (e.g., NASREQ for network access services). Re-use of existing AVPs is strongly recommended in [DIAMBASE].

For AVPs of type Enumerated, the service may require a new value to be defined. Allocation of new AVP values is done as specified in [DIAMBASE], section 1.2.

1b. New AVPs can be defined if the existing AVPs do not provide sufficient rating information. In this case, the procedures defined in [DIAMBASE] for creating new AVPs MUST be followed.

1c. For services specific only to one vendor's implementation, a Vendor-Specific AVP code for Private use can be used. Where a Vendor-Specific AVP is implemented by more than one vendor, allocation of global AVPs is encouraged instead; refer to [DIAMBASE].

2. The Service-Parameter-Info AVP MAY be used as a container to pass legacy rating information in its original encoded form (e.g., ASN.1 BER). This method can be used to avoid unnecessary conversions from an existing data format to an AVP format. In this case, the rating input is embedded in the Service-Parameter-Info AVP as defined in section 8.43.

New service applications SHOULD favor the use of explicitly defined AVPs as described in items 1a and 1b, to simplify interoperability.

4.1.1.2. Service-Specific Documentation

The service specific rating input AVPs, the contents of the Service-Parameter-Info AVP or Service-Context-Id AVP (defined in section 8.42) are not within the scope of this document. To facilitate interoperability, it is RECOMMENDED that the rating input and the values of the Service-Context-Id be coordinated via an informational RFC or other permanent and readily available reference. The specification of another cooperative standardization body (e.g., 3GPP, OMA, and 3GPP2) SHOULD be used. However, private services may be deployed that are subject to agreements between providers of the credit-control server and client. In this case, vendor specific AVPs can be used.

This specification, together with the above service specific documents, governs the credit-control message. Service specific documents define which existing AVPs or new AVPs are used as input to the rating process (i.e., those that do not define new credit-control applications), and thus have to be included in the Credit-Control-Request command by a Diameter credit-control client supporting a given service as *[AVP]. Should Service-Parameter-Info be used, then the service specific document MUST specify the exact content of this grouped AVP.

The Service-Context-Id AVP MUST be included at the command level of a Credit-Control Request to identify the service specific document that applies to the request. The specific service or rating group the request relates to is uniquely identified by the combination of Service-Context-Id and Service-Identifier or Rating-Group.

4.1.3. Handling of Unsupported/Incorrect Rating Input

Diameter credit-control implementations are required to support the Mandatory rating AVPs defined in service specific documentation of the services they support, according to the 'M' bit rules in [DIAMBASE].

If a rating input required for the rating process is incorrect in the Credit-control request, or if the credit-control server does not support the requested service context (identified by the Service-Context-Id AVP at command level), the Credit-control answer MUST contain the error code DIAMETER_RATING_FAILED. A CCA message with this error MUST contain one or more Failed-AVP AVPs containing the missing and/or unsupported AVPs that caused the failure. A Diameter credit-control client that receives the error code DIAMETER_RATING_FAILED in response to a request MUST NOT send similar requests in the future.

4.1.4. RADIUS Vendor-Specific Rating Attributes

When service specific documents include RADIUS vendor specific attributes that could be used as input in the rating process, the rules described in [NASREQ] for formatting the Diameter AVP MUST be followed.

For example, if the AVP code used is the vendor attribute type code, the Vendor-Specific flag MUST be set to 1 and the Vendor-ID MUST be set to the IANA Vendor identification value. The Diameter AVP data field contains only the attribute value of the RADIUS attribute.

5. Session Based Credit-Control

5.1. General Principles

For a session-based credit-control, several interrogations are needed: the first, intermediate (optional) and the final interrogations. This is illustrated in Figures 2 and 3.

If the credit-control client performs credit-reservation before granting service to the end user, it MUST use several interrogations toward the credit-control server (i.e., session based credit-

control). In this case, the credit-control server MUST maintain the credit-control session state.

Each credit-control session MUST have a globally unique Session-Id as defined in [DIAMBASE], which MUST NOT be changed during the lifetime of a credit-control session.

Certain applications require multiple credit-control sub-sessions. These applications would send messages with a constant Session-Id AVP, but with a different CC-Sub-Session-Id AVP. If several credit sub-sessions will be used, all sub-sessions MUST be closed separately before the main session is closed so that units per sub-session may be reported. The absence of this AVP implies that no sub-sessions are in use.

Note that the service element might send a service specific re-authorization message to the AAA server due to expiration of the authorization-lifetime during an ongoing credit-control session. However, the service specific re-authorization does not influence the credit authorization that is ongoing between the credit-control client and credit-control server, as credit authorization is controlled by the burning rate of the granted quota.

If service specific re-authorization fails, the user will be disconnected, and the credit-control client MUST send a final interrogation to the credit-control server.

The Diameter credit-control server may seek to control the validity time of the granted quota and/or the production of intermediate interrogations. Thus, it MAY include the Validity-Time AVP in the answer message to the credit-control client. Upon expiration of the Validity-Time, the credit-control client MUST generate a credit-control update request and report the used quota to the credit-control server. It is up to the credit-control server to determine the value of the Validity-Time to be used for consumption of the granted service units. If the Validity-Time is used, its value SHOULD be given as input to set the session supervision timer Tcc (the session supervision timer MAY be set to two times the value of the Validity-Time, as defined in section 13). Since credit-control update requests are also produced at the expiry of granted service units and/or for mid-session service events, the omission of Validity-Time does not mean that intermediate interrogation for the purpose of credit-control is not performed.

5.1.1. Basic Tariff-Time Change Support

The Diameter credit-control server and client MAY optionally support a tariff change mechanism. The Diameter credit-control server may include a Tariff-Time-Change AVP in the answer message. Note that the granted units should be allocated based on the worst-case scenario in case of forthcoming tariff change, so that the overall reported used units would never exceed the credit reservation.

When the Diameter credit-control client reports the used units and a tariff change has occurred during the reporting period, the Diameter credit-control client MUST separately itemize the units used before and after the tariff change. If the client is unable to distinguish whether units straddling the tariff change were used before or after the tariff change, the credit-control client MUST itemize those units in a third category.

If a client does not support the tariff change mechanism and it receives a CCA message carrying the Tariff-Time-Change AVP, it MUST terminate the credit-control session, giving a reason of DIAMETER_BAD_ANSWER in the Termination-Cause AVP.

For time based services, the quota is continuously consumed at the regular rate of 60 seconds per minute. At the time when credit resources are allocated, the server already knows how many units will be consumed before the tariff time change and how many units will be consumed afterward. Similarly, the server can determine the units consumed at the before rate and the units consumed at the rate afterward in the event that the end-user closes the session before the consumption of the allotted quota. There is no need for additional traffic between client and server in the case of tariff time changes for continuous time based service. Therefore, the tariff change mechanism is not used for such services. For time-based services in which the quota is NOT continuously consumed at a regular rate, the tariff change mechanism described for volume and event units MAY be used.

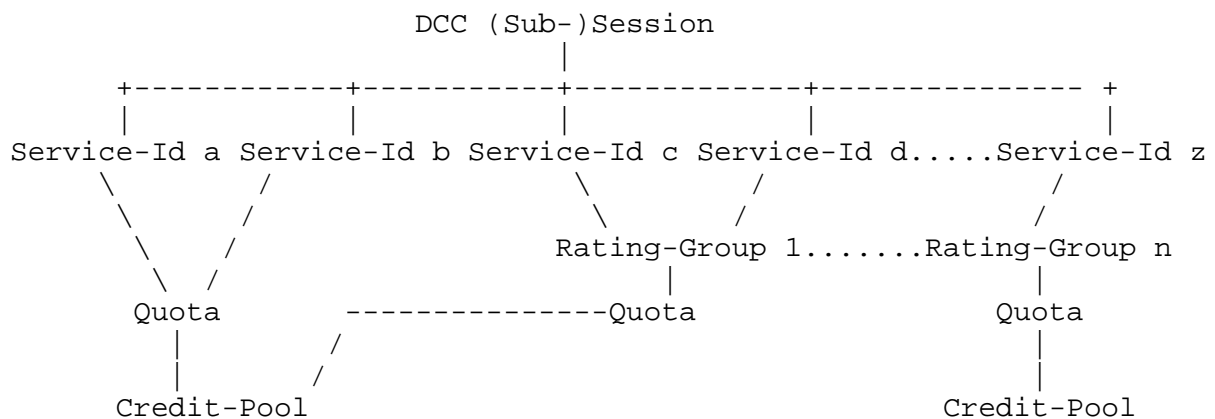
5.1.2. Credit-Control for Multiple Services within a (sub-)Session

When multiple services are used within the same user session and each service or group of services is subject to different cost, it is necessary to perform credit-control for each service independently. Making use of credit-control sub-sessions to achieve independent credit-control will result in increased signaling load and usage of resources in both the credit-control client and the credit-control server. For instance, during one network access session the end user may use several http-services subject to different access cost. The network access specific attributes such as the quality of service

(QoS) are common to all the services carried within the access bearer, but the cost of the bearer may vary depending on its content.

To support these scenarios optimally, the credit-control application enables independent credit-control of multiple services in a single credit-control (sub-)session. This is achieved by including the optional Multiple-Services-Credit-Control AVP in Credit-Control-Request/Answer messages. It is possible to request and allocate resources as a credit pool shared between multiple services. The services can be grouped into rating groups in order to achieve even further aggregation of credit allocation. It is also possible to request and allocate quotas on a per service basis. Where quotas are allocated to a pool by means of the Multiple-Services-Credit-Control AVP, the quotas remain independent objects that can be re-authorized independently at any time. Quotas can also be given independent result codes, validity times, and Final-Unit-Indications.

A Rating-Group gathers a set of services, identified by a Service-Identifier, and subject to the same cost and rating type (e.g., \$0.1/minute). It is assumed that the service element is provided with Rating-Groups, Service-Identifiers, and their associated parameters that define what has to be metered by means outside the scope of this specification. (Examples of parameters associated to Service-Identifiers are IP 5-tuple and HTTP URL.) Service-Identifiers enable authorization on a per-service based credit as well as itemized reporting of service usage. It is up to the credit-control server whether to authorize credit for one or more services or for the whole rating-group. However, the client SHOULD always report used units at the finest supported level of granularity. Where quota is allocated to a rating-group, all the services belonging to that group draw from the allotted quota. The following is a graphical representation of the relationship between service-identifiers, rating-groups, credit pools, and credit-control (sub-)session.



If independent credit-control of multiple services is used, the validity-time and final-unit-indication SHOULD be present either in the Multiple-Services-Credit-Control AVP(s) or at command level as single AVPs. However, the Result-Code AVP MAY be present both on the command level and within the Multiple-Services-Credit-Control AVP. If the Result-Code on the command level indicates a value other than SUCCESS, then the Result-Code on command level takes precedence over any included in the Multiple-Services-Credit-Control AVP.

The credit-control client MUST indicate support for independent credit-control of multiple services within a (sub-)session by including the Multiple-Services-Indicator AVP in the first interrogation. A credit-control server not supporting this feature MUST treat the Multiple-Services-Indicator AVP and any received Multiple-Services-Credit-Control AVPs as invalid AVPs.

If the client indicated support for independent credit-control of multiple services, a credit-control server that wishes to use the feature MUST return the granted units within the Multiple-Services-Credit-Control AVP associated to the corresponding service-identifier and/or rating-group.

To avoid a situation where several parallel (and typically also small) credit reservations must be made on the same account (i.e., credit fragmentation), and also to avoid unnecessary load on the credit-control server, it is possible to provide service units as a pool that applies to multiple services or rating groups. This is achieved by providing the service units in the form of a quota for a particular service or rating group in the Multiple-Services-Credit-Control AVP, and also by including a reference to a credit pool for that unit type.

The reference includes a multiplier derived from the rating parameter, which translates from service units of a specific type to the abstract service units in the pool. For instance, if the rating parameter for service 1 is \$1/MB and the rating parameter for service 2 is \$0.5/MB, the multipliers could be 10 and 5 for services 1 and 2, respectively.

If S is the total service units within the pool, M_1, M_2, \dots, M_n are the multipliers provided for services 1, 2, ..., n , and C_1, C_2, \dots, C_n are the used resources within the session, then the pool credit is exhausted and re-authorization MUST be sought when:

$$C_1 * M_1 + C_2 * M_2 + \dots + C_n * M_n \geq S$$

The total credit in the pool, S , is calculated from the quotas, which are currently allocated to the pool as follows:

$$S = Q1*M1 + Q2*M2 + \dots + Qn*Mn$$

If services or rating groups are added to or removed from the pool, then the total credit is adjusted appropriately. Note that when the total credit is adjusted because services or rating groups are removed from the pool, the value that need to be removed is the consumed one (i.e., $Cx*Mx$).

Re-authorizations for an individual service or rating group may be sought at any time; for example, if a 'non-pooled' quota is used up or the Validity-Time expires.

Where multiple G-S-U-Pool-Reference AVPs (section 8.30) with the same G-S-U-Pool-Identifier are provided within a Multiple-Services-Credit-Control AVP (section 8.16) along with the Granted-Service-Unit AVP, then these MUST have different CC-Unit-Type values, and they all draw from the credit pool separately. For instance, if one multiplier for time (Mlt) and one multiplier for volume (Mlv) are given, then the used resources from the pool is the sum $Clt*Mlt + Clv*Mlv$, where Clt is the time unit and Clv is the volume unit.

Where service units are provided within a Multiple-Services-Credit-Control AVP without a corresponding G-S-U-Pool-Reference AVP, then these are handled independently from any credit pool and from any other services or rating groups within the session.

The credit pool concept is an optimal tool to avoid the over-reservation effect of the basic single quota tariff time change mechanism (the mechanism described in section 5.1.1). Therefore, Diameter credit-control clients and servers implementing the independent credit-control of multiple services SHOULD leverage the credit pool concept when supporting the tariff time change. The Diameter credit-control server SHOULD include both the Tariff-Time-Change and Tariff-Change-Usage AVPs in two quota allocations in the answer message (i.e., two instances of the Multiple-Services-Credit-Control AVP). One of the granted units is allocated to be used before the potential tariff change, while the second granted units are for use after a tariff change. Both granted unit quotas MUST contain the same Service-Identifier and/or Rating-Group. This dual quota mechanism ensures that the overall reported used units would never exceed the credit reservation. The Diameter credit-control client reports both the used units before and after the tariff change in a single instance of the Multiple-Services-Credit-Control AVP.

The failure handling for credit-control sessions is defined in section 5.7 and reflected in the basic credit-control state machine in section 7. Credit-control clients and servers implementing the independent credit-control of multiple services in a (sub-)session functionality MUST ensure failure handling and general behavior fully consistent with the above mentioned sections, while maintaining the ability to handle parallel ongoing credit re-authorization within a (sub-)session. Therefore, it is RECOMMENDED that Diameter credit-control clients maintain a PendingU message queue and restart the Tx timer (section 13) every time a CCR message with the value UPDATE_REQUEST is sent while they are in PendingU state. When answers to all pending messages are received, the state machine moves to OPEN state, and Tx is stopped. Naturally, the action performed when a problem for the session is detected according to section 5.7 affects all the ongoing services (e.g., failover to a backup server if possible affect all the CCR messages with the value UPDATE_REQUEST in the PendingU queue).

Since the client may send CCR messages with the value UPDATE_REQUEST while in PendingU (i.e., without waiting for an answer to ongoing credit re-authorization), the time space between these requests may be very short, and the server may not have received the previous request(s) yet. Therefore, in this situation the server may receive out of sequence requests and SHOULD NOT consider this an error condition. A proper answer is to be returned to each of those requests.

5.2. First Interrogation

When session based credit-control is required (e.g., the authentication server indicated a prepaid user), the first interrogation MUST be sent before the Diameter credit-control client allows any service event to the end user. The CC-Request-Type is set to the value INITIAL_REQUEST in the request message.

If the Diameter credit-control client knows the cost of the service event (e.g., a content server delivering ringing tones may know their cost) the monetary amount to be charged is included in the Requested-Service-Unit AVP. If the Diameter credit-control client does not know the cost of the service event, the Requested-Service-Unit AVP MAY contain the number of requested service events. Where the Multiple-Services-Credit-Control AVP is used, it MUST contain the Requested-Service-Unit AVP to indicate that the quota for the associated service/rating-group is requested. In the case of multiple services, the Service-Identifier AVP or the Rating-Group AVP within the Multiple-Services-Credit-Control AVP always indicates the service concerned. Additional service event information to be rated

MAY be sent as service specific AVPs or MAY be sent within the Service-Parameter-Info AVP at command level. The Service-Context-Id AVP indicates the service specific document applicable to the request.

The Event-Timestamp AVP SHOULD be included in the request and contains the time when the service event is requested in the service element. The Subscription-Id AVP SHOULD be included to identify the end user in the credit-control server. The credit-control client MAY include the User-Equipment-Info AVP so that the credit-control server has some indication of the type and capabilities of the end user access device. How the credit-control server uses this information is outside the scope of this document.

The credit-control server SHOULD rate the service event and make a credit-reservation from the end user's account that covers the cost of the service event. If the type of the Requested-Service-Unit AVP is money, no rating is needed, but the corresponding monetary amount is reserved from the end user's account.

The credit-control server returns the Granted-Service-Unit AVP in the Answer message to the Diameter credit-control client. The Granted-Service-Unit AVP contains the amount of service units that the Diameter credit-control client can provide to the end user until a new Credit-Control-Request MUST be sent to the credit-control server. If several unit types are sent in the Answer message, the credit-control client MUST handle each unit type separately. The type of the Granted-Service-Unit AVP can be time, volume, service specific, or money, depending on the type of service event. The unit type(s) SHOULD NOT be changed within an ongoing credit-control session.

There MUST be a maximum of one instance of the same unit type in one Answer message. However, if multiple quotas are conveyed to the credit-control client in the Multiple-Services-Credit-Control AVPs, it is possible to carry two instances of the same unit type associated to a service-identifier/rating-group. This is typically the case when a tariff time change is expected and the credit-control server wants to make a distinction between the granted quota before and after tariff change.

If the credit-control server determines that no further control is needed for the service, it MAY include the result code indicating that the credit-control is not applicable (e.g., if the service is free of charge). This result code at command level implies that the credit-control session is to be terminated.

The Credit-Control-Answer message MAY also include the Final-Unit-Indication AVP to indicate that the answer message contains the final

units for the service. After the end user has consumed these units, the Diameter credit-control-client MUST behave as described in section 5.6.

This document defines two different approaches to perform the first interrogation to be used in different network architectures. The first approach uses credit-control messages after the user's authorization and authentication takes place. The second approach uses service specific authorization messages to perform the first interrogation during the user's authorization/authentication phase, and credit-control messages for the intermediate and final interrogations. If an implementation of the credit-control client supports both the methods, determining which method to use SHOULD be configurable.

In service environments such as the Network Access Server (NAS), it is desired to perform the first interrogation as part of the authorization/authentication process for the sake of protocol efficiency. Further credit authorizations after the first interrogation are performed with credit commands defined in this specification. Implementations of credit-control clients operating in the mentioned environments SHOULD support this method. If the credit-control server and AAA server are separate physical entities, the service element sends the request messages to the AAA server, which then issues an appropriate request or proxies the received request forward to the credit-control server.

In other service environments, such as the 3GPP network and some SIP scenarios, there is a substantial decoupling between registration/access to the network and the actual service request (i.e., the authentication/authorization is executed once at registration/access to the network and is not executed for every service event requested by the subscriber). In these environments, it is more appropriate to perform the first interrogation after the user has been authenticated and authorized. The first, the intermediate, and the final interrogations are executed with credit-control commands defined in this specification.

Other IETF standards or standards developed by other standardization bodies may define the most suitable method in their architectures.

5.2.1. First Interrogation after Authorization and Authentication

The Diameter credit-control client in the service element may get information from the authorization server as to whether credit-control is required, based on its knowledge of the end user. If credit-control is required the credit-control server needs to be contacted prior to initiating service delivery to the end user. The

accounting protocol and the credit-control protocol can be used in parallel. The authorization server may also determine whether the parallel accounting stream is required.

The following diagram illustrates the case where both protocols are used in parallel and the service element sends credit-control messages directly to the credit-control server. More credit-control sequence examples are given in Annex A.

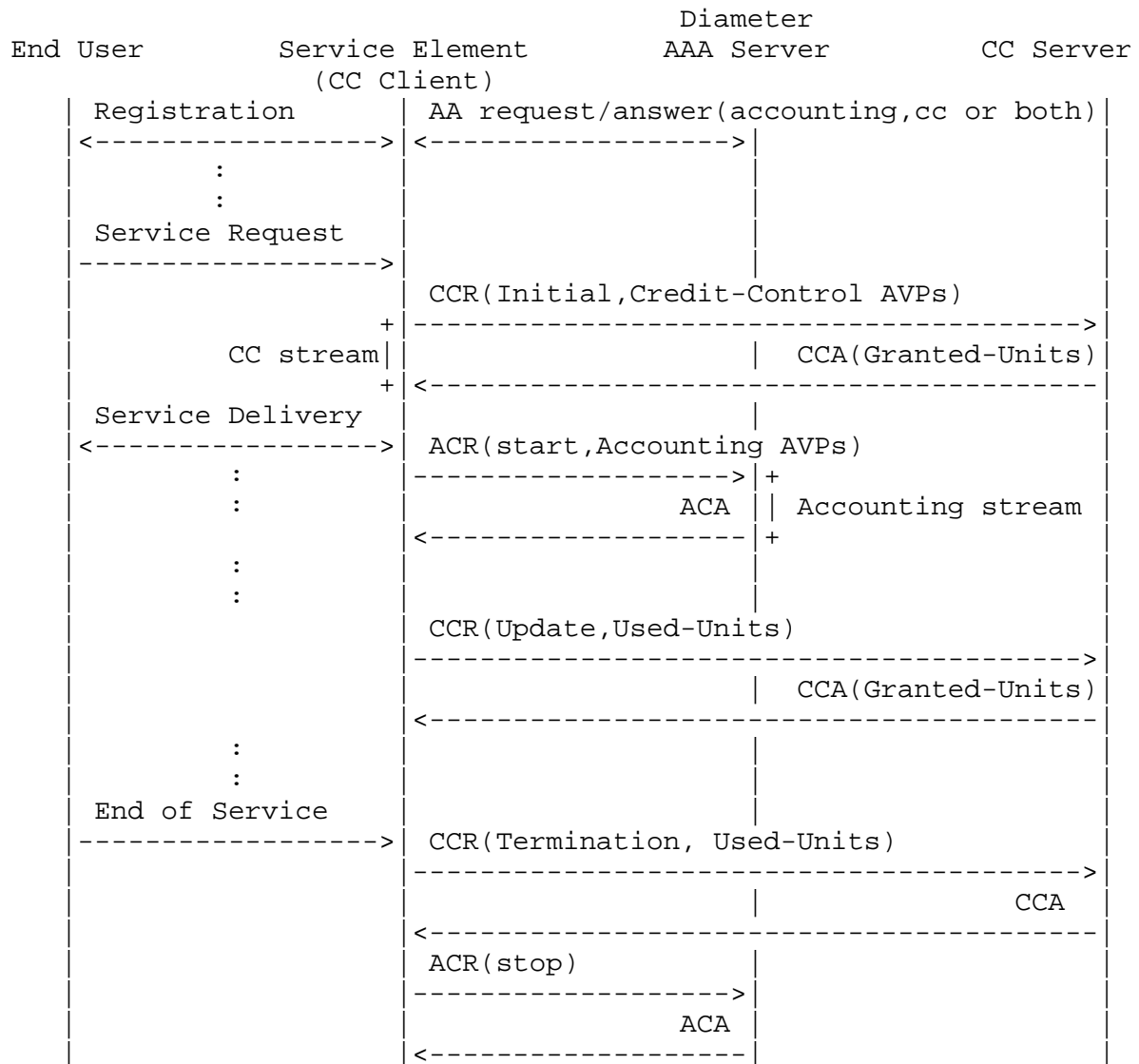


Figure 2: Protocol example with first interrogation after user's authorization/authentication

5.2.2. Authorization Messages for First Interrogation

The Diameter credit-control client in the service element MUST actively co-operate with the authorization/authentication client in the construction of the AA request by adding appropriate credit-control AVPs. The credit-control client MUST add the Credit-Control AVP to indicate credit-control capabilities and MAY add other relevant credit-control specific AVPs to the proper authorization/authentication command to perform the first interrogation toward the home Diameter AAA server. The Auth-Application-Id is set to the appropriate value, as defined in the relevant service specific authorization/authentication application document (e.g., [NASREQ], [DIAMMIP]). The home Diameter AAA server authenticates/authorizes the subscriber and determines whether credit-control is required.

If credit-control is not required for the subscriber, the home Diameter AAA server will respond as usual, with an appropriate AA answer message. If credit-control is required for the subscriber and the Credit-Control AVP with the value set to CREDIT_AUTHORIZATION was present in the authorization request, the home AAA server MUST contact the credit-control server to perform the first interrogation. If credit-control is required for the subscriber and the Credit-Control AVP was not present in the authorization request, the home AAA server MUST send an authorization reject answer message.

The Diameter AAA server supporting credit-control is required to send the Credit-Control-Request command (CCR) defined in this document to the credit-control server. The Diameter AAA server populates the CCR based on service specific AVPs used for input to the rating process, and possibly on credit-control AVPs received in the AA request. The credit-control server will reserve money from the user's account, will rate the request and will send a Credit-Control-Answer message to the home Diameter AAA server. The answer message includes the Granted-Service-Unit AVP(s) and MAY include other credit-control specific AVPs, as appropriate. Additionally, the credit-control server MAY set the Validity-Time and MAY include the Credit-Control-Failure-Handling AVP and the Direct-Debiting-Failure-Handling AVP to determine what to do if the sending of credit-control messages to the credit-control server has been temporarily prevented.

Upon receiving the Credit-Control-Answer message from the credit-control server, the home Diameter AAA server will populate the AA answer with the received credit-control AVPs and with the appropriate service attributes according to the authorization/authentication specific application (e.g., [NASREQ], [DIAMMIP]). It will then forward the packet to the credit-control client. If the home Diameter AAA server receives a credit-control reject message, it will

simply generate an appropriate authorization reject message to the credit-control client, including the credit-control specific error code.

In this model, the credit-control client sends further credit-control messages to the credit-control server via the home Diameter AAA server. Upon receiving a successful authorization answer message with the Granted-Service-Unit AVP(s), the credit-control client will grant the service to the end user and will generate an intermediate credit-control request, as required by using credit-control commands. The CC-Request-Number of the first UPDATE_REQUEST MUST be set to 1 (for how to produce unique value for the CC-Request-Number AVP, see section 8.2).

If service specific re-authorization is performed (i.e., authorization-lifetime expires), the credit-control client MUST add to the service specific re-authorization request the Credit-Control AVP with a value set to RE_AUTHORIZATION to indicate that the credit-control server MUST NOT be contacted. When session based credit-control is used for the subscriber, a constant credit-control message stream flows through the home Diameter AAA server. The home Diameter AAA server can make use of this credit-control message flow to deduce that the user's activity is ongoing; therefore, it is recommended to set the authorization-lifetime to a reasonably high value when credit-control is used for the subscriber.

In this scenario, the home Diameter AAA server MUST advertise support for the credit-control application to its peers during the capability exchange process.

The following diagram illustrates the use of authorization/authentication messages to perform the first interrogation. The parallel accounting stream is not shown in the figure.

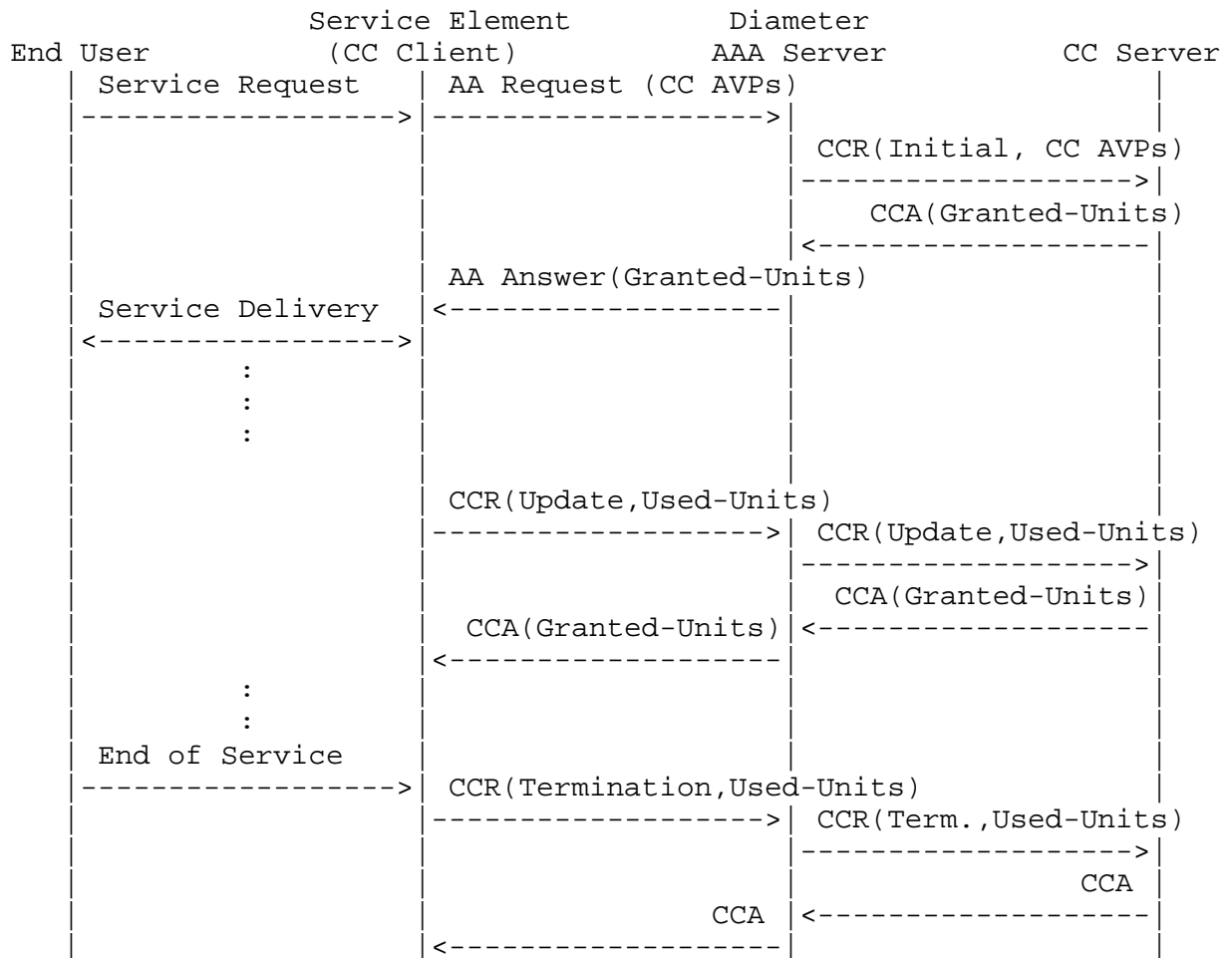


Figure 3: Protocol example with use of the authorization messages for the first interrogation

5.3. Intermediate Interrogation

When all the granted service units for one unit type are spent by the end user or the Validity-Time is expired, the Diameter credit-control client MUST send a new Credit-Control-Request to the credit-control server. In the event that credit-control for multiple services is applied in one credit-control session (i.e., units associated to Service-Identifier(s) or Rating-Group are granted), a new Credit-Control-Request MUST be sent to the credit-control server when the

credit reservation has been wholly consumed, or upon expiration of the Validity-Time. It is always up to the Diameter credit-control client to send a new request well in advance of the expiration of the previous request in order to avoid interruption in the service element. Even if the granted service units reserved by the credit-control server have not been spent upon expiration of the Validity-Time, the Diameter credit-control client MUST send a new Credit-Control-Request to the credit-control server.

There can also be mid-session service events, which might affect the rating of the current service events. In this case, a spontaneous updating (a new Credit-Control-Request) SHOULD be sent including information related to the service event even if all the granted service units have not been spent or the Validity-Time has not expired.

When the used units are reported to the credit-control server, the credit-control client will not have any units in its possession before new granted units are received from the credit-control server. When the new granted units are received, these units apply from the point where the measurement of the reported used units stopped. Where independent credit-control of multiple services is supported, this process may be executed for one or more services, a single rating-group, or a pool within the (sub)session.

The CC-Request-Type AVP is set to the value UPDATE_REQUEST in the intermediate request message. The Subscription-Id AVP SHOULD be included in the intermediate message to identify the end user in the credit-control server. The Service-Context-Id AVP indicates the service specific document applicable to the request.

The Requested-Service-Unit AVP MAY contain the new amount of requested service units. Where the Multiple-Services-Credit-Control AVP is used, it MUST contain the Requested-Service-Unit AVP if a new quota is requested for the associated service/rating-group. The Used-Service-Unit AVP contains the amount of used service units measured from the point when the service became active or, if interim interrogations are used during the session, from the point when the previous measurement ended. The same unit types used in the previous message SHOULD be used. If several unit types were included in the previous answer message, the used service units for each unit type MUST be reported.

The Event-Timestamp AVP SHOULD be included in the request and contains the time of the event that triggered the sending of the new Credit-Control-Request.

The credit-control server MUST deduct the used amount from the end user's account. It MAY rate the new request and make a new credit-reservation from the end user's account that covers the cost of the requested service event.

A Credit-Control-Answer message with the CC-Request-Type AVP set to the value UPDATE_REQUEST MAY include the Cost-Information AVP containing the accumulated cost estimation for the session, without taking any credit-reservation into account.

The Credit-Control-Answer message MAY also include the Final-Unit-Indication AVP to indicate that the answer message contains the final units for the service. After the end user has consumed these units, the Diameter credit-control-client MUST behave as described in section 5.6.

There can be several intermediate interrogations within a session.

5.4. Final Interrogation

When the end user terminates the service session, or when the graceful service termination described in section 5.6 takes place, the Diameter credit-control client MUST send a final Credit-Control-Request message to the credit-control server. The CC-Request-Type AVP is set to the value TERMINATION_REQUEST. The Service-Context-Id AVP indicates the service specific document applicable to the request.

The Event-Timestamp AVP SHOULD be included in the request and contains the time when the session was terminated.

The Used-Service-Unit AVP contains the amount of used service units measured from the point when the service became active or, if interim interrogations are used during the session, from the point when the previous measurement ended. If several unit types were included in the previous answer message, the used service units for each unit type MUST be reported.

After final interrogation, the credit-control server MUST refund the reserved credit amount not used to the end user's account and deduct the used monetary amount from the end user's account.

A Credit-Control-Answer message with the CC-Request-Type set to the value TERMINATION_REQUEST MAY include the Cost-Information AVP containing the estimated total cost for the session in question.

If the user logs off during an ongoing credit-control session, or if some other reason causes the user to become logged off (e.g., final-

unit indication causes user logoff according to local policy), the service element, according to application specific policy, may send a Session-Termination-Request (STR) to the home Diameter AAA server as usual [DIAMBASE]. Figure 4 illustrates the case when the final-unit indication causes user logoff upon consumption of the final granted units and the generation of STR.

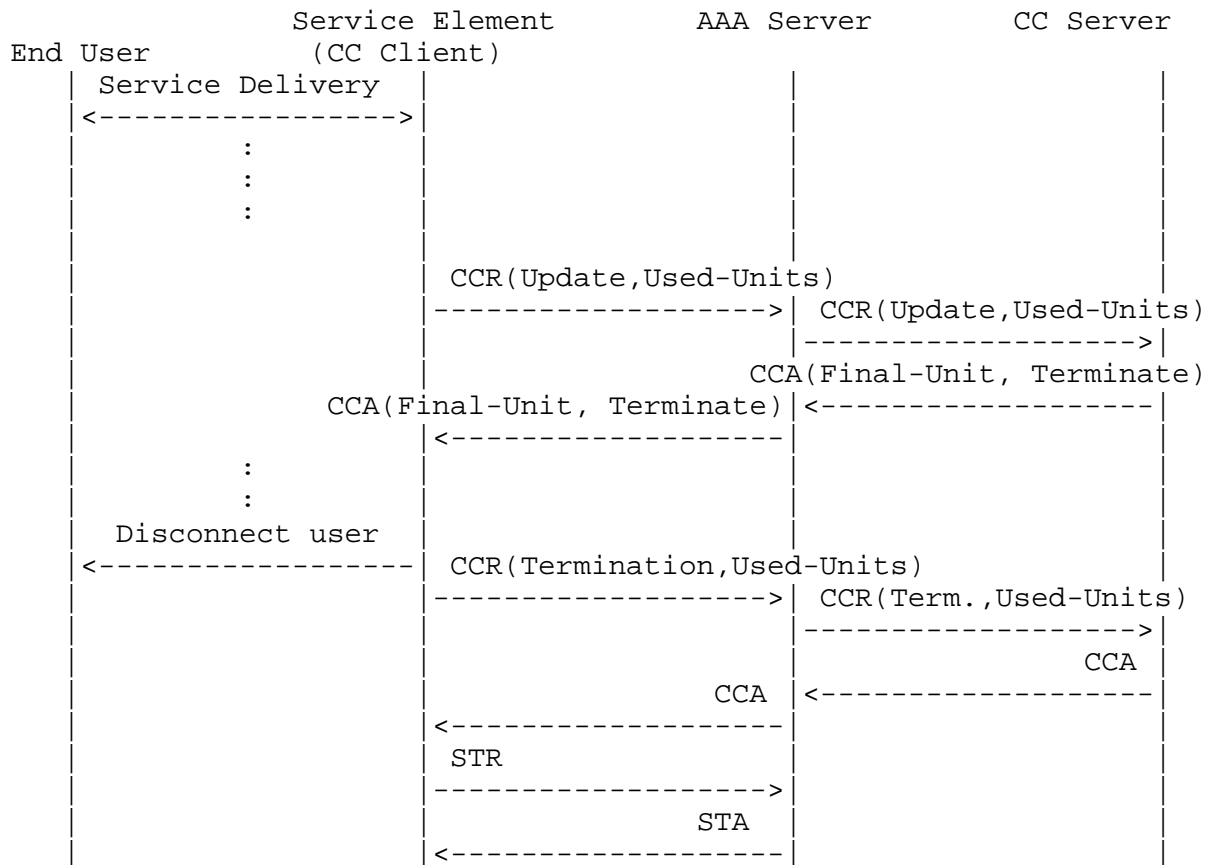


Figure 4: User disconnected due to exhausted account

5.5. Server-Initiated Credit Re-Authorization

The Diameter credit-control application supports server-initiated re-authorization. The credit-control server MAY optionally initiate the credit re-authorization by issuing a Re-Auth-Request (RAR) as defined in the Diameter base protocol [DIAMBASE]. The Auth-Application-Id in the RAR message is set to 4 to indicate Diameter Credit Control, and the Re-Auth-Request-Type is set to AUTHORIZE_ONLY.

Section 5.1.2 defines the feature to enable credit-control for multiple services within a single (sub-)session where the server can authorize credit usage at a different level of granularity. Further, the server may provide credit resources to multiple services or rating groups as a pool (see section 5.1.2 for details and definitions). Therefore, the server, based on its service logic and its knowledge of the ongoing session, can decide to request credit re-authorization for a whole (sub-)session, a single credit pool, a single service, or a single rating-group. To request credit re-authorization for a credit pool, the server includes in the RAR message the G-S-U-Pool-Identifier AVP indicating the affected pool. To request credit re-authorization for a service or a rating-group, the server includes in the RAR message the Service-Identifier AVP or the Rating-Group AVP, respectively. To request credit re-authorization for all the ongoing services within the (sub-)session, the server includes none of the above mentioned AVPs in the RAR message.

If a credit re-authorization is not already ongoing (i.e., the credit-control session is in Open state), a credit control client that receives an RAR message with Session-Id equal to a currently active credit-control session MUST acknowledge the request by sending the Re-Auth-Answer (RAA) message and MUST initiate the credit re-authorization toward the server by sending a Credit-Control-Request message with the CC-Request-Type AVP set to the value UPDATE_REQUEST. The Result-Code 2002 (DIAMETER_LIMITED_SUCCESS) SHOULD be used in the RAA message to indicate that an additional message (i.e., CCR message with the value UPDATE_REQUEST) is required to complete the procedure. If a quota was allocated to the service, the credit-control client MUST report the used quota in the Credit-Control-Request. Note that the end user does not need to be prompted for the credit re-authorization, since the credit re-authorization is transparent to the user (i.e., it takes place exclusively between the credit-control client and the credit-control server).

Where multiple services in a user's session are supported, the procedure in the above paragraph will be executed at the granularity requested by the server in the RAR message.

If credit re-authorization is ongoing at the time when the RAR message is received (i.e., RAR-CCR collision), the credit-control client successfully acknowledges the request but does not initiate a new credit re-authorization. The Result-Code 2001 (DIAMETER_SUCCESS) SHOULD be used in the RAA message to indicate that a credit re-authorization procedure is already ongoing (i.e., the client was in PendingU state when the RAR was received). The credit-control server SHOULD process the Credit-Control-Request as if it was received in answer to the server initiated credit re-authorization, and should

consider the server initiated credit re-authorization process successful upon reception of the Re-Auth-Answer message.

When multiple services are supported in a user's session, the server may request credit re-authorization for a credit pool (or for the (sub-)session) while a credit re-authorization is already ongoing for some of the services or rating-groups. In this case, the client acknowledges the server request with an RAA message and MUST send a new Credit-Control-Request message to perform re-authorization for the remaining services/rating-groups. The Result-Code 2002 (DIAMETER_LIMITED_SUCCESS) SHOULD be used in the RAA message to indicate that an additional message (i.e., CCR message with value UPDATE_REQUEST) is required to complete the procedure. The server processes the received requests and returns an appropriate answer to both requests.

The above-defined procedures are enabled for each of the possibly active Diameter credit-control sub-sessions. The server MAY request re-authorization for an active sub-session by including the CC-Sub-Session-Id AVP in the RAR message in addition to the Session-Id AVP.

5.6. Graceful Service Termination

When the user's account runs out of money, the user may not be allowed to compile additional chargeable events. However, the home service provider may offer some services; for instance, access to a service portal where it is possible to refill the account, for which the user is allowed to benefit for a limited time. The length of this time is usually dependent on the home service provider policy.

This section defines the optional graceful service termination feature that MAY be supported by the credit-control server. Credit-control client implementations MUST support the Final-Unit-Indication with at least the teardown of the ongoing service session once the subscriber has consumed all the final granted units.

Where independent credit-control of multiple services in a single credit-control (sub-)session is supported, it is possible to use the graceful service termination for each of the services/rating-groups independently. Naturally, the graceful service termination process defined in the following sub-sections will apply to the specific service/rating-group as requested by the server.

In some service environments (e.g., NAS), the graceful service termination may be used to redirect the subscriber to a service portal for online balance refill or other services offered by the home service provider. In this case, the graceful termination process installs a set of packet filters to restrict the user's

access capability only to/from the specified destinations. All the IP packets not matching the filters will be dropped or, possibly, re-directed to the service portal. The user may also be sent an appropriate notification as to why the access has been limited. These actions may be communicated explicitly from the server to the client or may be configured per-service at the client. Explicitly signaled redirect or restrict instructions always take precedence over configured ones.

It is also possible use the graceful service termination to connect the prepaid user to a top-up server that plays an announcement and prompts the user to replenish the account. In this case, the credit-control server sends only the address of the top-up server where the prepaid user shall be connected after the final granted units have been consumed. An example of this is given in Appendix A (Flow VII).

The credit-control server MAY initiate the graceful service termination by including the Final-Unit-Indication AVP in the Credit-Control-Answer to indicate that the message contains the final units for the service.

When the credit-control client receives the Final-Unit-Indication AVP in the answer from the server, its behavior depends on the value indicated in the Final-Unit-Action AVP. The server may request the following actions: TERMINATE, REDIRECT, or RESTRICT_ACCESS.

A following figure illustrates the graceful service termination procedure described in the following sub-sections.

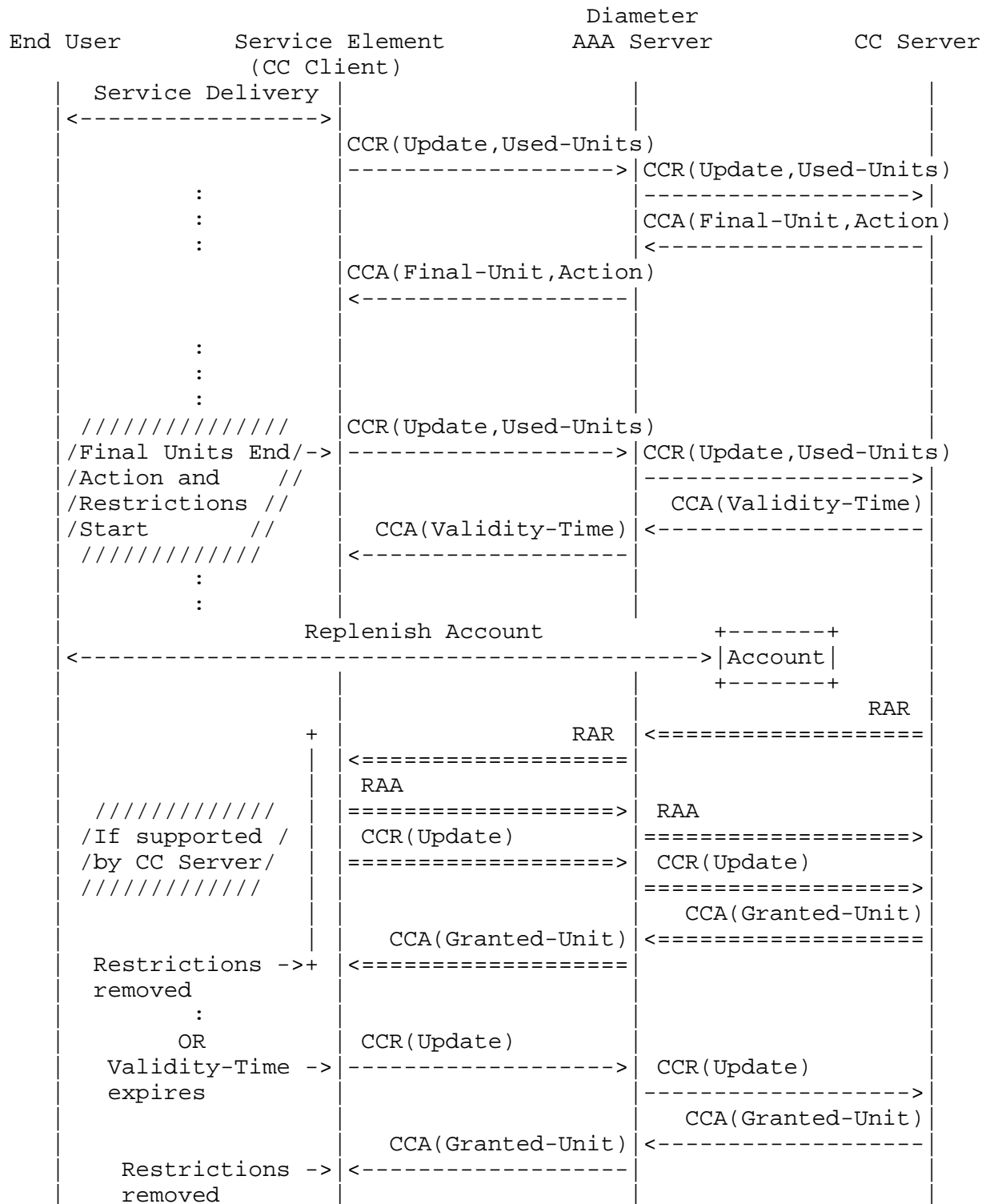


Figure 5: Optional graceful service termination procedure

5.6.1. Terminate Action

The Final-Unit-Indication AVP with Final-Unit-Action TERMINATE does not include any other information. When the subscriber has consumed the final granted units, the service element MUST terminate the service. This is the default handling applicable whenever the credit-control client receives an unsupported Final-Unit-Action value and MUST be supported by all the Diameter credit-control client implementations conforming to this specification. A final Credit-Control-Request message to the credit-control server MUST be sent if the Final-Unit-Indication AVP indicating action TERMINATE was present at command level. The CC-Request-Type AVP in the request is set to the value TERMINATION_REQUEST.

5.6.2. Redirect Action

The Final-Unit-Indication AVP with Final-Unit-Action REDIRECT indicates to the service element supporting this action that, upon consumption of the final granted units, the user MUST be re-directed to the address specified in the Redirect-Server AVP as follows.

The credit-control server sends the Redirect-Server AVP in the Credit-Control-Answer message. In such a case, the service element MUST redirect or connect the user to the destination specified in the Redirect-Server AVP, if possible. When the end user is redirected (by using protocols others than Diameter) to the specified server or connected to the top-up server, an additional authorization (and possibly authentication) may be needed before the subscriber can replenish the account; however, this is out of the scope of this specification.

In addition to the Redirect-Server AVP, the credit-control server MAY include one or more Restriction-Filter-Rule AVPs or one or more Filter-Id AVPs in the Credit-Control-Answer message to enable the user to access other services (for example, zero-rated services). In such a case, the access device MUST drop all the packets not matching the IP filters specified in the Credit-Control-Answer message and, if possible, redirect the user to the destination specified in the Redirect-Server AVP.

An entity other than the credit-control server may provision the access device with appropriate IP packet filters to be used in conjunction with the Diameter credit-control application. This case is considered in section 5.6.3.

When the final granted units have been consumed, the credit-control client MUST perform an intermediate interrogation. The purpose of this interrogation is to indicate to the credit-control server that the specified action started and to report the used units. The credit-control server MUST deduct the used amount from the end user's account but MUST NOT make a new credit reservation. The credit-control client, however, may send intermediate interrogations before all the final granted units have been consumed for which rating and money reservation may be needed; for instance, upon Validity-Time expires or upon mid-session service events that affect the rating of the current service. Therefore, the credit-control client MUST NOT include any rating related AVP in the request sent once all the final granted units have been consumed as an indication to the server that the requested final unit action started, rating and money reservation are not required (when the Multiple-Services-Credit-Control AVP is used, the Service-Identifier or Rating-Group AVPs is included to indicate the concerned services). Naturally, the Credit-Control-Answer message does not contain any granted service unit and MUST include the Validity-Time AVP to indicate to the credit-control client how long the subscriber is allowed to use network resources before a new intermediate interrogation is sent to the server.

At the expiry of Validity-Time, the credit-control client sends a Credit-Control-Request (UPDATE_REQUEST) as usual. This message does not include the Used-Service-Unit AVP, as there is no allotted quota to report. The credit-control server processes the request and MUST perform the credit reservation. If during this time the subscriber did not replenish his/her account, whether he/she will be disconnected or will be granted access to services not controlled by a credit-control server for an unlimited time is dependent on the home service provider policy (note: the latter option implies that the service element should not remove the restriction filters upon termination of the credit-control). The server will return the appropriate Result-Code (see section 9.1) in the Credit-Control-Answer message in order to implement the policy-defined action. Otherwise, new quota will be returned, the service element MUST remove all the possible restrictions activated by the graceful service termination process and continue the credit-control session and service session as usual.

The credit-control client may not wait until the expiration of the Validity-Time and may send a spontaneous update (a new Credit-Control-Request) if the service element can determine, for instance, that communication between the end user and the top-up server took place. An example of this is given in Appendix A (Figure A.8).

Note that the credit-control server may already have initiated the above-described process for the first interrogation. However, the user's account might be empty when this first interrogation is performed. In this case, the subscriber can be offered a chance to replenish the account and continue the service. The credit-control client receives a Credit-Control-Answer or service specific authorization answer with the Final-Unit-Indication and Validity-Time AVPs but no Granted-Service-Unit. It immediately starts the graceful service termination without sending any message to the server. An example of this case is illustrated in Appendix A.

5.6.3. Restrict Access Action

A Final-Unit-Indication AVP with the Final-Unit-Action `RESTRICT_ACCESS` indicates to the device supporting this action that the user's access **MUST** be restricted according to the IP packet filters given in the Restriction-Filter-Rule AVP(s) or according to the IP packet filters identified by the Filter-Id AVP(s). The credit-control server **SHOULD** include either the Restriction-Filter-Rule AVP or the Filter-Id AVP in the Credit-Control-Answer message.

An entity other than the credit-control server may provision the access device with appropriate IP packet filters to be used in conjunction with the Diameter credit-control application. Such an entity may, for instance, configure the access device with IP flows to be passed when the Diameter credit-control application indicates `RESTRICT_ACCESS` or `REDIRECT`. The access device passes IP packets according to the filter rules that may have been received in the Credit-Control-Answer message in addition to those that may have been configured by the other entity. However, when the user's account cannot cover the cost of the requested service, the action taken is the responsibility of the credit-control server that controls the prepaid subscriber.

If another entity working in conjunction with the Diameter credit-control application already provisions the access device with all the required filter rules for the end user, the credit-control server presumably need not send any additional filter. Therefore, it is **RECOMMENDED** that credit-control server implementations supporting the graceful service termination be configurable for sending the Restriction-Filter-Rule AVP, the Filter-Id AVP, or none of the above.

When the final granted units have been consumed, the credit-control client **MUST** perform an intermediate interrogation. The credit-control client and the credit-control server process this intermediate interrogation and execute subsequent procedures, as specified in the previous section for the `REDIRECT` action.

The credit-control server may initiate the graceful service termination with action `RESTRICT_ACCESS` already for the first interrogation, as specified in the previous section for the `REDIRECT` action.

5.6.4. Usage of the Server-Initiated Credit Re-Authorization

Once the subscriber replenishes the account, she presumably expects all the restrictions placed by the graceful termination procedure to be removed immediately and unlimited service' access to be resumed. For the best user experience, the credit-control server implementation MAY support the server-initiated credit re-authorization (see section 5.5). In such a case, upon the successful account top-up, the credit-control server sends the Re-Auth-Request (RAR) message to solicit the credit re-authorization. The credit-control client initiates the credit re-authorization by sending the Credit-Control-Request message with the CC-Request-Type AVP set to the value `UPDATE_REQUEST`. The Used-Service-Unit AVP is not included in the request, as there is no allotted quota to report. The Requested-Service-Unit AVP MAY be included in the request. After the credit-control client successfully receives the Credit-Control-Answer with new Granted-Service-Unit, all the possible restrictions activated for the purpose of the graceful service termination MUST be removed in the service element. The credit-control session and the service session continue as usual.

5.7. Failure Procedures

The Credit-Control-Failure-Handling AVP (CCFH), as described in this section, determines the behavior of the credit-control client in fault situations. The CCFH may be received from the Diameter home AAA server, from the credit-control server, or may be configured locally. The CCFH value received from the home AAA server overrides the locally configured value. The CCFH value received from the credit-control server in the Credit-Control-Answer message always overrides any existing value.

The authorization server MAY include the Accounting-Realtime-Required AVP to determine what to do if the sending of accounting records to the accounting server has been temporarily prevented, as defined in [DIAMBASE]. It is RECOMMENDED that the client complement the credit-control failure procedures with backup accounting flow toward an accounting server. By using different combinations of Accounting-Realtime-Required and Credit-Control-Failure-Handling AVPs, different safety levels can be built. For example, by choosing a Credit-Control-Failure-Handling AVP equal to `CONTINUE` for the credit-control flow and a Accounting-Realtime-Required AVP equal to `DELIVER_AND_GRANT` for the accounting flow, the service can be granted

to the end user even if the connection to the credit-control server is down, as long as the accounting server is able to collect the accounting information and information exchange is taking place between the accounting server and credit-control server.

As the credit-control application is based on real-time bi-directional communication between the credit-control client and the credit-control server, the usage of alternative destinations and the buffering of messages may not be sufficient in the event of communication failures. Because the credit-control server has to maintain session states, moving the credit-control message stream to a backup server requires a complex context transfer solution. Whether the credit-control message stream is moved to a backup credit-control server during an ongoing credit-control session depends on the value of the CC-Session-Failover AVP. However, failover may occur at any point in the path between the credit-control client and the credit-control server if a transport failure is detected with a peer, as described in [DIAMBASE]. As a consequence, the credit-control server might receive duplicate messages. These duplicates or out of sequence messages can be detected in the credit-control server based on the credit-control server session state machine (section 7), Session-Id AVP, and CC-Request-Number AVP.

If a failure occurs during an ongoing credit-control session, the credit-control client may move the credit-control message stream to an alternative server if the CC-server indicated `FAILOVER_SUPPORTED` in the CC-Session-Failover AVP. A secondary credit-control server name, either received from the home Diameter AAA server or configured locally, can be used as an address of the backup server. If the CC-Session-Failover AVP is set to `FAILOVER_NOT_SUPPORTED`, the credit-control message stream **MUST NOT** be moved to a backup server.

For new credit-control sessions, failover to an alternative credit-control server **SHOULD** be performed if possible. For instance, if an implementation of the credit-control client can determine primary credit-control server unavailability, it can establish the new credit-control sessions with a possibly available secondary credit-control server.

The AAA transport profile [AAATRANS] defines the application layer watchdog algorithm that enables failover from a peer that has failed and is controlled by a watchdog timer (*Tw*) defined in [AAATRANS]. The recommended default initial value for *Tw* (*Twinit*) is 30 seconds. *Twinit* may be set as low as 6 seconds; however, according to [AAATRANS], setting too low a value for *Twinit* is likely to result in an increased probability of duplicates, as well as an increase in spurious failover and fallback attempts. The Diameter base protocol

is common to several different types of Diameter AAA applications that may be run in the same service element. Therefore, tuning the timer Twinit to a lower value in order to satisfy the requirements of real-time applications, such as the Diameter credit-control application, will certainly cause the above mentioned problems. For prepaid services, however, the end user expects an answer from the network in a reasonable time. Thus, the Diameter credit-control client will react faster than would the underlying base protocol. Therefore this specification defines the timer Tx that is used by the credit-control client (as defined in section 13) to supervise the communication with the credit-control server. When the timer Tx elapses, the credit-control client takes an action to the end user according to the Credit-Control-Failure-Handling AVP.

When Tx expires, the Diameter credit-control client always terminates the service if the Credit-Control-Failure-Handling (CCFH) AVP is set to the value TERMINATE. The credit-control session may be moved to an alternative server only if a protocol error DIAMETER_TOO_BUSY or DIAMETER_UNABLE_TO_DELIVER is received before Tx expires. Therefore, the value TERMINATE is not appropriate if proper failover behavior is desired.

If the Credit-Control-Failure-Handling AVP is set to the value CONTINUE or RETRY_AND_TERMINATE, the service will be granted to the end user when the timer Tx expires. An answer message with granted-units may arrive later if the base protocol transport failover occurred in the path to the credit-control server. (The Twinit default value is 3 times more than the Tx recommended value.) The credit-control client SHOULD grant the service to the end user, start monitoring the resource usage, and wait for the possible late answer until the timeout of the request (e.g., 120 seconds). If the request fails and the CC-Session-Failover AVP is set to FAILOVER_NOT_SUPPORTED, the credit-control client terminates or continues the service depending on the value set in the CCFH and MUST free all the reserved resources for the credit-control session. If the protocol error DIAMETER_UNABLE_TO_DELIVER or DIAMETER_TOO_BUSY is received or the request times out and the CC-Session-Failover AVP is set to FAILOVER_SUPPORTED, the credit-control client MAY send the request to a backup server, if possible. If the credit-control client receives a successful answer from the backup server, it continues the credit-control session with such a server. If the re-transmitted request also fails, the credit-control client terminates or continues the service depending on the value set in the CCFH and MUST free all the reserved resources for the credit-control session.

If a communication failure occurs during the graceful service termination procedure, the service element SHOULD always terminate the ongoing service session.

If the credit-control server detects a failure during an ongoing credit-control session, it will terminate the credit-control session and return the reserved units back to the end user's account.

The supervision session timer Tcc (as defined in section 13) is used in the credit-control server to supervise the credit-control session.

In order to support failover between credit-control servers, information transfer about the credit-control session and account state SHOULD take place between the primary and the secondary credit-control server. Implementations supporting the credit-control session failover MUST also ensure proper detection of duplicate or out of sequence messages. The communication between the servers is regarded as an implementation issue and is outside of the scope of this specification.

6. One Time Event

The one-time event is used when there is no need to maintain any state in the Diameter credit-control server; for example, enquiring about the price of the service. The use of a one-time event implies that the user has been authenticated and authorized beforehand.

The one time event can be used when the credit-control client wants to know the cost of the service event or to check the account balance without any credit-reservation. It can also be used for refunding service units on the user's account or for direct debiting without any credit-reservation. The one time event is shown in Figure 6.

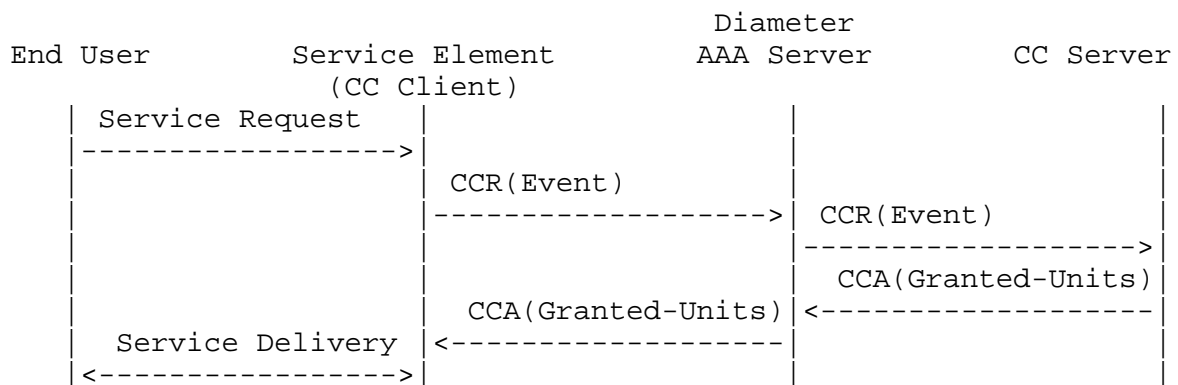


Figure 6: One time event

In environments such as the 3GPP architecture, the one time event can be sent from the service element directly to the credit-control server.

6.1. Service Price Enquiry

The credit-control client may need to know the price of the service event. Services offered by application service providers whose prices are not known in the credit-control client might exist. The end user might also want to get an estimation of the price of a service event before requesting it.

A Diameter credit-control client requesting the cost information MUST set the CC-Request-Type AVP equal to EVENT_REQUEST, include the Requested-Action AVP set to PRICE_ENQUIRY, and set the requested service event information into the Service-Identifier AVP in the Credit-Control-Request message. Additional service event information may be sent as service specific AVPs or within the Service-Parameter-Info AVP. The Service-Context-Id AVP indicates the service specific document applicable to the request.

The credit-control server calculates the cost of the requested service event, but it does not perform any account balance check or credit-reservation from the account.

The estimated cost of the requested service event is returned to the credit-control client in the Cost-Information AVP in the Credit-Control-Answer message.

6.2. Balance Check

The Diameter credit-control client may only have to verify that the end user's account balance covers the cost of a certain service without reserving any units from the account at the time of the inquiry. This method does not guarantee that credit would be left when the Diameter credit-control client requests the debiting of the account with a separate request.

A Diameter credit-control client requesting the balance check MUST set the CC-Request-Type AVP equal to EVENT_REQUEST, include a Requested-Action AVP set to CHECK_BALANCE, and include the Subscription-Id AVP in order to identify the end user in the credit-control server. The Service-Context-Id AVP indicates the service specific document applicable to the request.

The credit-control server makes the balance check, but it does not make any credit-reservation from the account.

The result of balance check (ENOUGH_CREDIT/NO_CREDIT) is returned to the credit-control client in the Check-Balance-Result AVP in the Credit-Control-Answer message.

6.3. Direct Debiting

There are certain service events for which service execution is always successful in the service environment. The delay between the service invocation and the actual service delivery to the end user can be sufficiently long that the use of the session-based credit-control would lead to unreasonably long credit-control sessions. In these cases, the Diameter credit-control client can use the one-time event scenario for direct debiting. The Diameter credit-control client SHOULD be sure that the requested service event execution would be successful when this scenario is used.

In the Credit-Control-Request message, the CC-Request-Type is set to the value EVENT_REQUEST and the Requested-Action AVP is set to DIRECT_DEBITING. The Subscription-Id AVP SHOULD be included to identify the end user in the credit-control server. The Event-Timestamp AVP SHOULD be included in the request and contain the time when the service event is requested in the service element. The Service-Context-Id AVP indicates the service specific document applicable to the request.

The Diameter credit-control client MAY include the monetary amount to be charged in the Requested-Service-Unit AVP, if it knows the cost of the service event. If the Diameter credit-control client does not know the cost of the service event, the Requested-Service-Unit AVP MAY contain the number of requested service events. The Service-Identifier AVP always indicates the service concerned. Additional service event information to be rated MAY be sent as service specific AVPs or within the Service-Parameter-Info AVP.

The credit-control server SHOULD rate the service event and deduct the corresponding monetary amount from the end user's account. If the type of the Requested-Service-Unit AVP is money, no rating is needed, but the corresponding monetary amount is deducted from the end user's account.

The credit-control server returns the Granted-Service-Unit AVP in the Credit-Control-Answer message to the Diameter credit-control client. The Granted-Service-Unit AVP contains the amount of service units that the Diameter credit-control client can provide to the end user. The type of the Granted-Service-Unit can be time, volume, service specific, or money, depending on the type of service event.

If the credit-control server determines that no credit-control is needed for the service, it can include the result code indicating that the credit-control is not applicable (e.g., service is free of charge).

For informative purposes, the Credit-Control-Answer message MAY also include the Cost-Information AVP containing the estimated total cost of the requested service.

6.4. Refund

Some services may refund service units to the end user's account; for example, gaming services.

The credit-control client MUST set CC-Request-Type to the value EVENT_REQUEST and the Requested-Action AVP to REFUND_ACCOUNT in the Credit-Control-Request message. The Subscription-Id AVP SHOULD be included to identify the end user in the credit-control server. The Service-Context-Id AVP indicates the service specific document applicable to the request.

The Diameter credit-control client MAY include the monetary amount to be refunded in the Requested-Service-Unit AVP. The Service-Identifier AVP always indicates the concerned service. If the Diameter credit-control client does not know the monetary amount to be refunded, in addition to the Service-Identifier AVP it MAY send service specific AVPs or the Service-Parameter-Info AVP containing additional service event information to be rated.

For informative purposes, the Credit-Control-Answer message MAY also include the Cost-Information AVP containing the estimated monetary amount of refunded unit.

6.5. Failure Procedure

Failover to an alternative credit-control server is allowed for a one time event, as the server is not maintaining session states. For instance, if the credit-control client receives a protocol error DIAMETER_UNABLE_TO_DELIVER or DIAMETER_TOO_BUSY, it can re-send the request to an alternative server, if possible. There MAY be protocol transparent Diameter relays and redirect agents or Diameter credit-control proxies between the credit-control client and credit-control server. Failover may occur at any point in the path between the credit-control client and the credit-control server if a transport failure is detected with a peer, as described in [DIAMBASE]. Because there can be duplicate requests for various reasons, the credit-control server is responsible for real time duplicate detection. Implementation issues for duplicate detection are discussed in [DIAMBASE], Appendix C.

When the credit-control client detects a communication failure with the credit-control server, its behavior depends on the requested action. The timer Tx (as defined in section 13) is used in the

credit-control client to supervise the communication with the credit-control server.

If the requested action is PRICE_ENQUIRY or CHECK_BALANCE and communication failure is detected, the credit-control client SHOULD forward the request messages to an alternative credit-control server, if possible. The secondary credit-control server name, if received from the home Diameter AAA server, can be used as an address of backup server.

If the requested action is DIRECT_DEBITING, the Direct-Debiting-Failure-Handling AVP (DDFH) controls the credit-control client's behavior. The DDFH may be received from the home Diameter AAA server or may be locally configured. The credit-control server may also send the DDFH in any CCA message to be used for direct debiting events compiled thereafter. The DDFH value received from the home Diameter AAA server overrides the locally configured value, and the DDFH value received from the credit-control server in a Credit-Control-Answer message always overrides any existing value.

If the DDFH is set to TERMINATE_OR_BUFFER, the credit-control client SHOULD NOT grant the service if it can determine, eventually after a possible re-transmission attempt to an alternative credit-control server, from the result code or error code in the answer message that units have not been debited. Otherwise, the credit-control client SHOULD grant the service to the end user and store the request in the credit-control application level non-volatile storage. (Note that re-sending the request at a later time is not a guarantee that the service will be debited, as the user's account may be empty when the server successfully processes the request.) The credit-control client MUST mark these request messages as possible duplicates by setting the T-flag in the command header as described in [DIAMBASE], section 3.

If the Direct-Debiting-Failure-Handling AVP is set to CONTINUE, the service SHOULD be granted, even if credit-control messages cannot be delivered and messages are not buffered.

If the timer Tx expires, the credit-control client MUST continue the service and wait for a possible late answer. If the request times out, the credit-control client re-transmits the request (marked with T-flag) to a backup credit-control server, if possible. If the re-transmitted request also times out, or if a temporary error is received in answer, the credit-control client buffers the request if the value of the Direct-Debiting-Failure-Handling AVP is set to TERMINATE_OR_BUFFER. If a failed answer is received for the

re-transmitted request, the credit-control client frees all the resources reserved for the event message and deletes the request regardless of the value of the DDFH.

The Credit-Control-Request with the requested action REFUND_ACCOUNT should always be stored in the credit-control application level non-volatile storage in case of temporary failure. The credit-control client MUST mark the re-transmitted request message as a possible duplicate by setting the T-flag in the command header as described in [DIAMBASE], section 3.

For stored requests, the implementation may choose to limit the number of re-transmission attempts and to define a re-transmission interval.

Note that only one place in the credit-control system SHOULD be responsible for duplicate detection. If there is only one credit-control server within the given realm, the credit-control server may perform duplicate detection. If there is more than one credit-control server in a given realm, only one entity in the credit-control system should be responsible, to ensure that the end user's account is not debited or credited multiple times for the same service event.

7. Credit-Control Application State Machine

This section defines the credit-control application state machine.

The first four state machines are to be observed by credit-control clients. The first one describes the session-based credit-control when the first interrogation is executed as part of the authorization/authentication process. The second describes the session-based credit-control when the first interrogation is executed after the authorization/authentication process. The requirements as to what state machines have to be supported are discussed in section 5.2.

The third state machine describes the session-based credit-control for the intermediate and final interrogations. The fourth one describes the event-based credit-control. These latter state machines are to be observed by all implementations that conform to this specification.

The fifth state machine describes the credit-control session from a credit-control server perspective.

Any event not listed in the state machines MUST be considered an error condition, and a corresponding answer, if applicable, MUST be returned to the originator of the message.

In the state table, the event 'Failure to send' means that the Diameter credit-control client is unable to communicate with the desired destination or, if failover procedure is supported, with a possibly defined alternative destination (e.g., the request times out and the answer message is not received). This could be due to the peer being down, or due to a physical link failure in the path to or from the credit-control server.

The event 'Temporary error' means that the Diameter credit-control client received a protocol error notification (DIAMETER_TOO_BUSY, DIAMETER_UNABLE_TO_DELIVER, or DIAMETER_LOOP_DETECTED) in the Result-Code AVP of the Credit-Control-Answer command. The above protocol error notification may ultimately be received in answer to the re-transmitted request to a defined alternative destination, if failover is supported.

The event 'Failed answer' means that the Diameter credit-control client received non-transient failure (permanent failure) notification in the Credit-Control-Answer command. The above permanent failure notification may ultimately be received in answer to the re-transmitted request to a defined alternative destination, if failover is supported.

The action 'store request' means that a request is stored in the credit-control application level non-volatile storage.

The event 'Not successfully processed' means that the credit-control server could not process the message; e.g., due to an unknown end user, account being empty, or errors defined in [DIAMBASE].

The event 'User service terminated' can be triggered by various reasons, e.g., normal user termination, network failure, and ASR (Abort-Session-Request). The Termination-Cause AVP contains information about the termination reason, as specified in [DIAMBASE].

The Tx timer, which is used to control the waiting time in the credit-control client in the Pending state, is stopped upon exit of the Pending state. The stopping of the Tx timer is omitted in the state machine when the new state is Idle, as moving to Idle state implies the clearing of the session and all the variables associated to it.

The states PendingI, PendingU, PendingT, PendingE, and PendingB stand for pending states to wait for an answer to a credit-control request related to Initial, Update, Termination, Event, or Buffered request, respectively.

The acronyms CCFH and DDFH stand for Credit-Control-Failure-Handling and Direct-Debiting-Failure-Handling, respectively.

In the following state machine table, the failover to a secondary server upon 'Temporary error' or 'Failure to send' is not explicitly described. Moving an ongoing credit-control message stream to an alternative server is, however, possible if the CC-Session-Failover AVP is set to FAILOVER_SUPPORTED, as described in section 5.7.

Re-sending a credit-control event to an alternative server is supported as described in section 6.5.

CLIENT, SESSION BASED for the first interrogation with AA request

| State | Event | Action | New State |
|----------|--|--|-----------|
| Idle | Client or device requests access/service | Send AA request with added CC AVPs, start Tx | PendingI |
| PendingI | Successful AA req. answer received | Grant service to end user, stop Tx | Open |
| PendingI | Tx expired | Disconnect user/dev | Idle |
| PendingI | Failed AA answer received | Disconnect user/dev | Idle |
| PendingI | AA answer received with result code equal to CREDIT_CONTROL_NOT_APPLICABLE | Grant service to end user | Idle |
| PendingI | User service terminated | Queue termination event | PendingI |

| | | | |
|----------|----------------------------|--------------------------------------|----------|
| PendingI | Change in rating condition | Queue changed rating condition event | PendingI |
|----------|----------------------------|--------------------------------------|----------|

CLIENT, SESSION BASED for the first interrogation with CCR

| State | Event | Action | New State |
|----------|---|--------------------------------|-----------|
| ----- | | | |
| Idle | Client or device requests access/service | Send CC initial req., start Tx | PendingI |
| PendingI | Successful CC initial answer received | Stop Tx | Open |
| PendingI | Failure to send, or temporary error and CCFH equal to CONTINUE | Grant service to end user | Idle |
| PendingI | Failure to send, or temporary error and CCFH equal to TERMINATE or to RETRY_AND_TERMINATE | Terminate end user's service | Idle |
| PendingI | Tx expired and CCFH equal to TERMINATE | Terminate end user's service | Idle |
| PendingI | Tx expired and CCFH equal to CONTINUE or to RETRY_AND_TERMINATE | Grant service to end user | PendingI |
| PendingI | CC initial answer received with result code END_USER_SERVICE_DENIED or USER_UNKNOWN | Terminate end user's service | Idle |
| PendingI | CC initial answer received with result code equal to CREDIT_CONTROL_NOT_APPLICABLE | Grant service to end user | Idle |

| | | | |
|----------|---|--------------------------------------|----------|
| PendingI | Failed CC initial answer received and CCFH equal to CONTINUE | Grant service to end user | Idle |
| PendingI | Failed CC initial answer received and CCFH equal to TERMINATE or to RETRY_AND_TERMINATE | Terminate end user's service | Idle |
| PendingI | User service terminated | Queue termination event | PendingI |
| PendingI | Change in rating condition | Queue changed rating condition event | PendingI |

CLIENT, SESSION BASED for intermediate and final interrogations

| State | Event | Action | New State |
|-------|--|--|-----------|
| ----- | | | |
| Open | Granted unit elapses and no final unit indication received | Send CC update req., start Tx | PendingU |
| Open | Granted unit elapses and final unit action equal to TERMINATE received | Terminate end user's service, send CC termination req. | PendingT |
| Open | Change in rating condition in queue | Send CC update req., Start Tx | PendingU |
| Open | Service terminated in queue | Send CC termination req. | PendingT |
| Open | Change in rating condition or Validity-Time elapses | Send CC update req., Start Tx | PendingU |

| | | | |
|----------|---|---|----------|
| Open | User service terminated | Send CC termination req. | PendingT |
| Open | RAR received | Send RAA followed by CC update req., start Tx | PendingU |
| PendingU | Successful CC update answer received | Stop Tx | Open |
| PendingU | Failure to send, or temporary error and CCFH equal to CONTINUE | Grant service to end user | Idle |
| PendingU | Failure to send, or temporary error and CCFH equal to TERMINATE or to RETRY_AND_TERMINATE | Terminate end user's service | Idle |
| PendingU | Tx expired and CCFH equal to TERMINATE | Terminate end user's service | Idle |
| PendingU | Tx expired and CCFH equal to CONTINUE or to RETRY_AND_TERMINATE | Grant service to end user | PendingU |
| PendingU | CC update answer received with result code END_USER_SERVICE_DENIED | Terminate end user's service | Idle |
| PendingU | CC update answer received with result code equal to CREDIT_CONTROL_NOT_APPLICABLE | Grant service to end user | Idle |
| PendingU | Failed CC update answer received and CCFH equal to CONTINUE | Grant service to end user | Idle |
| PendingU | Failed CC update answer received and CCFH equal to TERMINATE or to RETRY_AND_TERMINATE | Terminate end user's service | Idle |

| | | | |
|----------|--|--------------------------------------|----------|
| PendingU | User service terminated | Queue termination event | PendingU |
| PendingU | Change in rating condition | Queue changed rating condition event | PendingU |
| PendingU | RAR received | Send RAA | PendingU |
| PendingT | Successful CC termination answer received | | Idle |
| PendingT | Failure to send, temporary error, or failed answer | | Idle |
| PendingT | Change in rating condition | | PendingT |

CLIENT, EVENT BASED

| State | Event | Action | New State |
|----------|---|------------------------------|-----------|
| Idle | Client or device requests a one-time service | Send CC event req., Start Tx | PendingE |
| Idle | Request in storage | Send stored request | PendingB |
| PendingE | Successful CC event answer received | Grant service to end user | Idle |
| PendingE | Failure to send, temporary error, failed CC event answer received, or Tx expired; requested action CHECK_BALANCE or PRICE_ENQUIRY | Indicate service error | Idle |
| PendingE | CC event answer received with result code END_USER_SERVICE_DENIED or USER_UNKNOWN and Tx running | Terminate end user's service | Idle |

| | | | |
|----------|---|---|----------|
| PendingE | CC event answer received with result code CREDIT_CONTROL_NOT_APPLICABLE; requested action DIRECT_DEBITING | Grant service to end user | Idle |
| PendingE | Failure to send, temporary error, or failed CC event answer received; requested action DIRECT_DEBITING; DDFH equal to CONTINUE | Grant service to end user | Idle |
| PendingE | Failed CC event answer received or temporary error; requested action DIRECT_DEBITING; DDFH equal to TERMINATE_OR_BUFFER and Tx running | Terminate end user's service | Idle |
| PendingE | Tx expired; requested action DIRECT_DEBITING | Grant service to end user | PendingE |
| PendingE | Failure to send; requested action DIRECT_DEBITING; DDFH equal to TERMINATE_OR_BUFFER | Store request with T-flag | Idle |
| PendingE | Temporary error; requested action DIRECT_DEBITING; DDFH equal to TERMINATE_OR_BUFFER; Tx expired | Store request | Idle |
| PendingE | Failed answer or answer received with result code END_USER_SERVICE_DENIED or USER_UNKNOWN; requested action DIRECT_DEBITING; Tx expired | | Idle |
| PendingE | Failed CC event answer received; requested action REFUND_ACCOUNT | Indicate service error and delete request | Idle |

| | | | |
|----------|--|---------------------------|------|
| PendingE | Failure to send or Tx expired; requested action REFUND_ACCOUNT | Store request with T-flag | Idle |
| PendingE | Temporary error, and requested action REFUND_ACCOUNT | Store request | Idle |
| PendingB | Successful CC answer received | Delete request | Idle |
| PendingB | Failed CC answer received | Delete request | Idle |
| PendingB | Failure to send or temporary error | | Idle |

SERVER, SESSION AND EVENT BASED

| State | Event | Action | New State |
|-------|--|--|-----------|
| ----- | | | |
| Idle | CC initial request received and successfully processed | Send CC initial answer, reserve units, start Tcc | Open |
| Idle | CC initial request received but not successfully processed | Send CC initial answer with Result-Code != SUCCESS | Idle |
| Idle | CC event request received and successfully processed | Send CC event answer | Idle |
| Idle | CC event request received but not successfully processed | Send CC event answer with Result-Code != SUCCESS | Idle |

| | | | |
|------|--|---|------|
| Open | CC update request received and successfully processed | Send CC update answer, debit used units, reserve new units, restart Tcc | Open |
| Open | CC update request received but not successfully processed | Send CC update answer with Result-Code != SUCCESS, debit used units | Idle |
| Open | CC termination request received and successfully processed | Send CC termin. answer, Stop Tcc, debit used units | Idle |
| Open | CC termination request received but not successfully processed | Send CC termin. answer with Result-Code != SUCCESS, debit used units | Idle |
| Open | Session supervision timer Tcc expired | Release reserved units | Idle |

8. Credit-Control AVPs

This section defines the credit-control AVPs that are specific to Diameter credit-control application and that MAY be included in the Diameter credit-control messages.

The AVPs defined in this section MAY also be included in authorization commands defined in authorization-specific applications, such as [NASREQ] and [DIAMMIP], if the first interrogation is performed as part of the authorization/authentication process, as described in section 5.2.

The Diameter AVP rules are defined in the Diameter Base [DIAMBASE], section 4. These AVP rules are observed in AVPs defined in this section.

The following table describes the Diameter AVPs defined in the credit-control application, their AVP Code values, types, possible flag values, and whether the AVP MAY be encrypted. The Diameter base [DIAMBASE] specifies the AVP Flag rules for AVPs in section 4.5.

| Attribute Name | AVP Code | Section Defined | Data Type | AVP Flag rules | | | | Encr |
|----------------------------------|----------|-----------------|-------------|----------------|-----|----------|----------|------|
| | | | | MUST | MAY | SHLD NOT | MUST NOT | |
| CC-Correlation-Id | 411 | 8.1 | OctetString | | P,M | | V | Y |
| CC-Input-Octets | 412 | 8.24 | Unsigned64 | M | P | | V | Y |
| CC-Money | 413 | 8.22 | Grouped | M | P | | V | Y |
| CC-Output-Octets | 414 | 8.25 | Unsigned64 | M | P | | V | Y |
| CC-Request-Number | 415 | 8.2 | Unsigned32 | M | P | | V | Y |
| CC-Request-Type | 416 | 8.3 | Enumerated | M | P | | V | Y |
| CC-Service-Specific-Units | 417 | 8.26 | Unsigned64 | M | P | | V | Y |
| CC-Session-Failover | 418 | 8.4 | Enumerated | M | P | | V | Y |
| CC-Sub-Session-Id | 419 | 8.5 | Unsigned64 | M | P | | V | Y |
| CC-Time | 420 | 8.21 | Unsigned32 | M | P | | V | Y |
| CC-Total-Octets | 421 | 8.23 | Unsigned64 | M | P | | V | Y |
| CC-Unit-Type | 454 | 8.32 | Enumerated | M | P | | V | Y |
| Check-Balance-Result | 422 | 8.6 | Enumerated | M | P | | V | Y |
| Cost-Information | 423 | 8.7 | Grouped | M | P | | V | Y |
| Cost-Unit | 424 | 8.12 | UTF8String | M | P | | V | Y |
| Credit-Control | 426 | 8.13 | Enumerated | M | P | | V | Y |
| Credit-Control-Failure-Handling | 427 | 8.14 | Enumerated | M | P | | V | Y |
| Currency-Code | 425 | 8.11 | Unsigned32 | M | P | | V | Y |
| Direct-Debiting-Failure-Handling | 428 | 8.15 | Enumerated | M | P | | V | Y |
| Exponent | 429 | 8.9 | Integer32 | M | P | | V | Y |
| Final-Unit-Action | 449 | 8.35 | Enumerated | M | P | | V | Y |
| Final-Unit-Indication | 430 | 8.34 | Grouped | M | P | | V | Y |
| Granted-Service-Unit | 431 | 8.17 | Grouped | M | P | | V | Y |
| G-S-U-Pool-Identifier | 453 | 8.31 | Unsigned32 | M | P | | V | Y |

| | | | | | | | | |
|----------------------------------|-----|------|--------------|---|-----|--|---|---|
| G-S-U-Pool-Reference | 457 | 8.30 | Grouped | M | P | | V | Y |
| Multiple-Services-Credit-Control | 456 | 8.16 | Grouped | M | P | | V | Y |
| Multiple-Services-Indicator | 455 | 8.40 | Enumerated | M | P | | V | Y |
| Rating-Group | 432 | 8.29 | Unsigned32 | M | P | | V | Y |
| Redirect-Address-Type | 433 | 8.38 | Enumerated | M | P | | V | Y |
| Redirect-Server | 434 | 8.37 | Grouped | M | P | | V | Y |
| Redirect-Server-Address | 435 | 8.39 | UTF8String | M | P | | V | Y |
| Requested-Action | 436 | 8.41 | Enumerated | M | P | | V | Y |
| Requested-Service-Unit | 437 | 8.18 | Grouped | M | P | | V | Y |
| Restriction-Filter-Rule | 438 | 8.36 | IPFilterRule | M | P | | V | Y |
| Service-Context-Id | 461 | 8.42 | UTF8String | M | P | | V | Y |
| Service-Identifier | 439 | 8.28 | Unsigned32 | M | P | | V | Y |
| Service-Parameter-Info | 440 | 8.43 | Grouped | | P,M | | V | Y |
| Service-Parameter-Type | 441 | 8.44 | Unsigned32 | | P,M | | V | Y |
| Service-Parameter-Value | 442 | 8.45 | OctetString | | P,M | | V | Y |
| Subscription-Id | 443 | 8.46 | Grouped | M | P | | V | Y |
| Subscription-Id-Data | 444 | 8.48 | UTF8String | M | P | | V | Y |
| Subscription-Id-Type | 450 | 8.47 | Enumerated | M | P | | V | Y |
| Tariff-Change-Usage | 452 | 8.27 | Enumerated | M | P | | V | Y |
| Tariff-Time-Change | 451 | 8.20 | Time | M | P | | V | Y |
| Unit-Value | 445 | 8.8 | Grouped | M | P | | V | Y |
| Used-Service-Unit | 446 | 8.19 | Grouped | M | P | | V | Y |
| User-Equipment-Info | 458 | 8.49 | Grouped | | P,M | | V | Y |
| User-Equipment-Info-Type | 459 | 8.50 | Enumerated | | P,M | | V | Y |
| User-Equipment-Info-Value | 460 | 8.51 | OctetString | | P,M | | V | Y |
| Value-Digits | 447 | 8.10 | Integer64 | M | P | | V | Y |
| Validity-Time | 448 | 8.33 | Unsigned32 | M | P | | V | Y |

8.1. CC-Correlation-Id AVP

The CC-Correlation-Id AVP (AVP Code 411) is of type OctetString and contains information to correlate credit-control requests generated for different components of the service; e.g., transport and service level. The one who allocates the Service-Context-Id (i.e., unique identifier of a service specific document) is also responsible for defining the content and encoding of the CC-Correlation-Id AVP.

8.2. CC-Request-Number AVP

The CC-Request-Number AVP (AVP Code 415) is of type Unsigned32 and identifies this request within one session. As Session-Id AVPs are globally unique, the combination of Session-Id and CC-Request-Number AVPs is also globally unique and can be used in matching credit-control messages with confirmations. An easy way to produce unique numbers is to set the value to 0 for a credit-control request of type INITIAL_REQUEST and EVENT_REQUEST and to set the value to 1 for the first UPDATE_REQUEST, to 2 for the second, and so on until the value for TERMINATION_REQUEST is one more than for the last UPDATE_REQUEST.

8.3. CC-Request-Type AVP

The CC-Request-Type AVP (AVP Code 416) is of type Enumerated and contains the reason for sending the credit-control request message. It MUST be present in all Credit-Control-Request messages. The following values are defined for the CC-Request-Type AVP:

- | | |
|---|---|
| INITIAL_REQUEST | 1 |
| An Initial request is used to initiate a credit-control session, and contains credit control information that is relevant to the initiation. | |
| UPDATE_REQUEST | 2 |
| An Update request contains credit-control information for an existing credit-control session. Update credit-control requests SHOULD be sent every time a credit-control re-authorization is needed at the expiry of the allocated quota or validity time. Further, additional service-specific events MAY trigger a spontaneous Update request. | |
| TERMINATION_REQUEST | 3 |
| A Termination request is sent to terminate a credit-control session and contains credit-control information relevant to the existing session. | |

EVENT_REQUEST

4

An Event request is used when there is no need to maintain any credit-control session state in the credit-control server. This request contains all information relevant to the service, and is the only request of the service. The reason for the Event request is further detailed in the Requested-Action AVP. The Requested-Action AVP MUST be included in the Credit-Control-Request message when CC-Request-Type is set to EVENT_REQUEST.

8.4. CC-Session-Failover AVP

The CC-Session-Failover AVP (AVP Code 418) is type of Enumerated and contains information as to whether moving the credit-control message stream to a backup server during an ongoing credit-control session is supported. In communication failures, the credit-control message streams can be moved to an alternative destination if the credit-control server supports failover to an alternative server. The secondary credit-control server name, if received from the home Diameter AAA server, can be used as an address of the backup server. An implementation is not required to support moving a credit-control message stream to an alternative server, as this also requires moving information related to the credit-control session to backup server.

The following values are defined for the CC-Session-Failover AVP:

FAILOVER_NOT_SUPPORTED

0

When the CC-Session-Failover AVP is set to FAILOVER_NOT_SUPPORTED, the credit-control message stream MUST NOT to be moved to an alternative destination in the case of communication failure.

This is the default behavior if the AVP isn't included in the reply from the authorization or credit-control server.

FAILOVER_SUPPORTED

1

When the CC-Session-Failover AVP is set to FAILOVER_SUPPORTED, the credit-control message stream SHOULD be moved to an alternative destination in the case of communication failure. Moving the credit-control message stream to a backup server MAY require that information related to the credit-control session should also be forwarded to alternative server.

8.5. CC-Sub-Session-Id AVP

The CC-Sub-Session-Id AVP (AVP Code 419) is of type Unsigned64 and contains the credit-control sub-session identifier. The combination of the Session-Id and this AVP MUST be unique per sub-session, and

the value of this AVP MUST be monotonically increased by one for all new sub-sessions. The absence of this AVP implies that no sub-sessions are in use.

8.6. Check-Balance-Result AVP

The Check Balance Result AVP (AVP Code 422) is of type Enumerated and contains the result of the balance check. This AVP is applicable only when the Requested-Action AVP indicates CHECK_BALANCE in the Credit-Control-Request command.

The following values are defined for the Check-Balance-Result AVP.

```
ENOUGH_CREDIT          0
  There is enough credit in the account to cover the requested
  service.
```

```
NO_CREDIT          1
  There isn't enough credit in the account to cover the requested
  service.
```

8.7. Cost-Information AVP

The Cost-Information AVP (AVP Code 423) is of type Grouped, and it is used to return the cost information of a service, which the credit-control client can transfer transparently to the end user. The included Unit-Value AVP contains the cost estimate (always type of money) of the service, in the case of price enquiry, or the accumulated cost estimation, in the case of credit-control session.

The Currency-Code specifies in which currency the cost was given. The Cost-Unit specifies the unit when the service cost is a cost per unit (e.g., cost for the service is \$1 per minute).

When the Requested-Action AVP with value PRICE_ENQUIRY is included in the Credit-Control-Request command, the Cost-Information AVP sent in the succeeding Credit-Control-Answer command contains the cost estimation of the requested service, without any reservation being made.

The Cost-Information AVP included in the Credit-Control-Answer command with the CC-Request-Type set to UPDATE_REQUEST contains the accumulated cost estimation for the session, without taking any credit reservation into account.

The Cost-Information AVP included in the Credit-Control-Answer command with the CC-Request-Type set to EVENT_REQUEST or TERMINATION_REQUEST contains the estimated total cost for the requested service.

It is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Cost-Information ::= < AVP Header: 423 >
                    { Unit-Value }
                    { Currency-Code }
                    [ Cost-Unit ]
```

8.8. Unit-Value AVP

Unit-Value AVP is of type Grouped (AVP Code 445) and specifies the units as decimal value. The Unit-Value is a value with an exponent; i.e., $\text{Unit-Value} = \text{Value-Digits AVP} * 10^{\text{Exponent}}$. This representation avoids unwanted rounding off. For example, the value of 2,3 is represented as Value-Digits = 23 and Exponent = -1. The absence of the exponent part MUST be interpreted as an exponent equal to zero.

It is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Unit-Value ::= < AVP Header: 445 >
               { Value-Digits }
               [ Exponent ]
```

8.9. Exponent AVP

Exponent AVP is of type Integer32 (AVP Code 429) and contains the exponent value to be applied for the Value-Digit AVP within the Unit-Value AVP.

8.10. Value-Digits AVP

The Value-Digits AVP is of type Integer64 (AVP Code 447) and contains the significant digits of the number. If decimal values are needed to present the units, the scaling MUST be indicated with the related Exponent AVP. For example, for the monetary amount \$ 0.05 the value of Value-Digits AVP MUST be set to 5, and the scaling MUST be indicated with the Exponent AVP set to -2.

8.11. Currency-Code AVP

The Currency-Code AVP (AVP Code 425) is of type Unsigned32 and contains a currency code that specifies in which currency the values of AVPs containing monetary units were given. It is specified by using the numeric values defined in the ISO 4217 standard [ISO4217].

8.12. Cost-Unit AVP

The Cost-Unit AVP (AVP Code 424) is of type UTF8String, and it is used to display a human readable string to the end user. It specifies the applicable unit to the Cost-Information when the service cost is a cost per unit (e.g., cost of the service is \$1 per minute). The Cost-Unit can be minutes, hours, days, kilobytes, megabytes, etc.

8.13. Credit-Control AVP

The Credit-Control AVP (AVP Code 426) is of type Enumerated and MUST be included in AA requests when the service element has credit-control capabilities.

CREDIT_AUTHORIZATION 0

If the home Diameter AAA server determines that the user has prepaid subscription, this value indicates that the credit-control server MUST be contacted to perform the first interrogation. The value of the Credit-Control AVP MUST always be set to 0 in an AA request sent to perform the first interrogation and to initiate a new credit-control session.

RE_AUTHORIZATION 1

This value indicates to the Diameter AAA server that a credit-control session is ongoing for the subscriber and that the credit-control server MUST not be contacted. The Credit-Control AVP set to the value of 1 is to be used only when the first interrogation has been successfully performed and the credit-control session is ongoing (i.e., re-authorization triggered by Authorization-Lifetime). This value MUST NOT be used in an AA request sent to perform the first interrogation.

8.14. Credit-Control-Failure-Handling AVP

The Credit-Control-Failure-Handling AVP (AVP Code 427) is of type Enumerated. The credit-control client uses information in this AVP to decide what to do if sending credit-control messages to the credit-control server has been, for instance, temporarily prevented due to a network problem. Depending on the service logic, the credit-control server can order the client to terminate the service

immediately when there is a reason to believe that the service cannot be charged, or to try failover to an alternative server, if possible. Then the server could either terminate or grant the service, should the alternative connection also fail.

TERMINATE

0

When the Credit-Control-Failure-Handling AVP is set to TERMINATE, the service MUST only be granted for as long as there is a connection to the credit-control server. If the credit-control client does not receive any Credit-Control-Answer message within the Tx timer (as defined in section 13), the credit-control request is regarded as failed, and the end user's service session is terminated.

This is the default behavior if the AVP isn't included in the reply from the authorization or credit-control server.

CONTINUE

1

When the Credit-Control-Failure-Handling AVP is set to CONTINUE, the credit-control client SHOULD re-send the request to an alternative server in the case of transport or temporary failures, provided that a failover procedure is supported in the credit-control server and the credit-control client, and that an alternative server is available. Otherwise, the service SHOULD be granted, even if credit-control messages can't be delivered.

RETRY_AND_TERMINATE

2

When the Credit-Control-Failure-Handling AVP is set to RETRY_AND_TERMINATE, the credit-control client SHOULD re-send the request to an alternative server in the case of transport or temporary failures, provided that a failover procedure is supported in the credit-control server and the credit-control client, and that an alternative server is available. Otherwise, the service SHOULD not be granted when the credit-control messages can't be delivered.

8.15. Direct-Debiting-Failure-Handling AVP

The Direct-Debiting-Failure-Handling AVP (AVP Code 428) is of type Enumerated. The credit-control client uses information in this AVP to decide what to do if sending credit-control messages (Requested-Action AVP set to DIRECT_DEBITING) to the credit-control server has been, for instance, temporarily prevented due to a network problem.

TERMINATE_OR_BUFFER

0

When the Direct-Debiting-Failure-Handling AVP is set to TERMINATE_OR_BUFFER, the service MUST be granted for as long as there is a connection to the credit-control server. If the

credit-control client does not receive any Credit-Control-Answer message within the Tx timer (as defined in section 13) the credit-control request is regarded as failed. The client SHOULD terminate the service if it can determine from the failed answer that units have not been debited. Otherwise the credit-control client SHOULD grant the service, store the request in application level non-volatile storage, and try to re-send the request. These requests MUST be marked as possible duplicates by setting the T-flag in the command header as described in [DIAMBASE] section 3.

This is the default behavior if the AVP isn't included in the reply from the authorization server.

CONTINUE

1

When the Direct-Debiting-Failure-Handling AVP is set to CONTINUE, the service SHOULD be granted, even if credit-control messages can't be delivered, and the request should be deleted.

8.16. Multiple-Services-Credit-Control AVP

Multiple-Services-Credit-Control AVP (AVP Code 456) is of type Grouped and contains the AVPs related to the independent credit-control of multiple services feature. Note that each instance of this AVP carries units related to one or more services or related to a single rating group.

The Service-Identifier and the Rating-Group AVPs are used to associate the granted units to a given service or rating group. If both the Service-Identifier and the Rating-Group AVPs are included, the target of the service units is always the service(s) indicated by the value of the Service-Identifier AVP(s). If only the Rating-Group-Id AVP is present, the Multiple-Services-Credit-Control AVP relates to all the services that belong to the specified rating group.

The G-S-U-Pool-Reference AVP allows the server to specify a G-S-U-Pool-Identifier identifying a credit pool within which the units of the specified type are considered pooled. If a G-S-U-Pool-Reference AVP is present, then actual service units of the specified type MUST also be present. For example, if the G-S-U-Pool-Reference AVP specifies Unit-Type TIME, then the CC-Time AVP MUST be present.

The Requested-Service-Unit AVP MAY contain the amount of requested service units or the requested monetary value. It MUST be present in the initial interrogation and within the intermediate interrogations in which new quota is requested. If the credit-control client does not include the Requested-Service-Unit AVP in a request command, because for instance, it has determined that the end-user terminated

the service, the server MUST debit the used amount from the user's account but MUST NOT return a new quota in the corresponding answer. The Validity-Time, Result-Code, and Final-Unit-Indication AVPs MAY be present in an answer command as defined in sections 5.1.2 and 5.6 for the graceful service termination.

When both the Tariff-Time-Change and Tariff-Change-Usage AVPs are present, the server MUST include two separate instances of the Multiple-Services-Credit-Control AVP with the Granted-Service-Unit AVP associated to the same service-identifier and/or rating-group. Where the two quotas are associated to the same pool or to different pools, the credit pooling mechanism defined in section 5.1.2 applies. The Tariff-Change-Usage AVP MUST NOT be included in request commands to report used units before, and after tariff time change the Used-Service-Unit AVP MUST be used.

A server not implementing the independent credit-control of multiple services functionality MUST treat the Multiple-Services-Credit-Control AVP as an invalid AVP.

The Multiple-Services-Control AVP is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Multiple-Services-Credit-Control ::= < AVP Header: 456 >
    [ Granted-Service-Unit ]
    [ Requested-Service-Unit ]
    *[ Used-Service-Unit ]
    [ Tariff-Change-Usage ]
    *[ Service-Identifier ]
    [ Rating-Group ]
    *[ G-S-U-Pool-Reference ]
    [ Validity-Time ]
    [ Result-Code ]
    [ Final-Unit-Indication ]
    *[ AVP ]
```

8.17. Granted-Service-Unit AVP

Granted-Service-Unit AVP (AVP Code 431) is of type Grouped and contains the amount of units that the Diameter credit-control client can provide to the end user until the service must be released or the new Credit-Control-Request must be sent. A client is not required to implement all the unit types, and it must treat unknown or unsupported unit types in the answer message as an incorrect CCA answer. In this case, the client MUST terminate the credit-control session and indicate in the Termination-Cause AVP reason DIAMETER_BAD_ANSWER.

The Granted-Service-Unit AVP is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Granted-Service-Unit ::= < AVP Header: 431 >
    [ Tariff-Time-Change ]
    [ CC-Time ]
    [ CC-Money ]
    [ CC-Total-Octets ]
    [ CC-Input-Octets ]
    [ CC-Output-Octets ]
    [ CC-Service-Specific-Units ]
    *[ AVP ]
```

8.18. Requested-Service-Unit AVP

The Requested-Service-Unit AVP (AVP Code 437) is of type Grouped and contains the amount of requested units specified by the Diameter credit-control client. A server is not required to implement all the unit types, and it must treat unknown or unsupported unit types as invalid AVPs.

The Requested-Service-Unit AVP is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Requested-Service-Unit ::= < AVP Header: 437 >
    [ CC-Time ]
    [ CC-Money ]
    [ CC-Total-Octets ]
    [ CC-Input-Octets ]
    [ CC-Output-Octets ]
    [ CC-Service-Specific-Units ]
    *[ AVP ]
```

8.19. Used-Service-Unit AVP

The Used-Service-Unit AVP is of type Grouped (AVP Code 446) and contains the amount of used units measured from the point when the service became active or, if interim interrogations are used during the session, from the point when the previous measurement ended.

The Used-Service-Unit AVP is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Used-Service-Unit ::= < AVP Header: 446 >
                        [ Tariff-Change-Usage ]
                        [ CC-Time ]
                        [ CC-Money ]
                        [ CC-Total-Octets ]
                        [ CC-Input-Octets ]
                        [ CC-Output-Octets ]
                        [ CC-Service-Specific-Units ]
                        *[ AVP ]
```

8.20. Tariff-Time-Change AVP

The Tariff-Time-Change AVP (AVP Code 451) is of type Time. It is sent from the server to the client and includes the time in seconds since January 1, 1900, 00:00 UTC, when the tariff of the service will be changed.

The tariff change mechanism is optional for the client and server, and it is not used for time-based services defined in section 5. If a client does not support the tariff time change mechanism, it MUST treat Tariff-Time-Change AVP in the answer message as an incorrect CCA answer. In this case, the client terminates the credit-control session and indicates in the Termination-Cause AVP reason DIAMETER_BAD_ANSWER.

Omission of this AVP means that no tariff change is to be reported.

8.21. CC-Time AVP

The CC-Time AVP (AVP Code 420) is of type Unsigned32 and indicates the length of the requested, granted, or used time in seconds.

8.22. CC-Money AVP

The CC-Money AVP (AVP Code 413) is of type Grouped and specifies the monetary amount in the given currency. The Currency-Code AVP SHOULD be included. It is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
CC-Money ::= < AVP Header: 413 >
              { Unit-Value }
              [ Currency-Code ]
```

8.23. CC-Total-Octets AVP

The CC-Total-Octets AVP (AVP Code 421) is of type Unsigned64 and contains the total number of requested, granted, or used octets regardless of the direction (sent or received).

8.24. CC-Input-Octets AVP

The CC-Input-Octets AVP (AVP Code 412) is of type Unsigned64 and contains the number of requested, granted, or used octets that can be/have been received from the end user.

8.25. CC-Output-Octets AVP

The CC-Output-Octets AVP (AVP Code 414) is of type Unsigned64 and contains the number of requested, granted, or used octets that can be/have been sent to the end user.

8.26. CC-Service-Specific-Units AVP

The CC-Service-Specific-Units AVP (AVP Code 417) is of type Unsigned64 and specifies the number of service-specific units (e.g., number of events, points) given in a selected service. The service-specific units always refer to the service identified in the Service-Identifier AVP (or Rating-Group AVP when the Multiple-Services-Credit-Control AVP is used).

8.27. Tariff-Change-Usage AVP

The Tariff-Change-Usage AVP (AVP Code 452) is of type Enumerated and defines whether units are used before or after a tariff change, or whether the units straddled a tariff change during the reporting period. Omission of this AVP means that no tariff change has occurred.

In addition, when present in answer messages as part of the Multiple-Services-Credit-Control AVP, this AVP defines whether units are allocated to be used before or after a tariff change event.

When the Tariff-Time-Change AVP is present, omission of this AVP in answer messages means that the single quota mechanism applies.

Tariff-Change-Usage can be one of the following:

UNIT_BEFORE_TARIFF_CHANGE 0

When present in the Multiple-Services-Credit-Control AVP, this value indicates the amount of the units allocated for use before a tariff change occurs.

When present in the Used-Service-Unit AVP, this value indicates the amount of resource units used before a tariff change had occurred.

UNIT_AFTER_TARIFF_CHANGE 1

When present in the Multiple-Services-Credit-Control AVP, this value indicates the amount of the units allocated for use after a tariff change occurs.

When present in the Used-Service-Unit AVP, this value indicates the amount of resource units used after tariff change had occurred.

UNIT_INDETERMINATE 2

The used unit contains the amount of units that straddle the tariff change (e.g., the metering process reports to the credit-control client in blocks of *n* octets, and one block straddled the tariff change). This value is to be used only in the Used-Service-Unit AVP.

8.28. Service-Identifier AVP

The Service-Identifier AVP is of type Unsigned32 (AVP Code 439) and contains the identifier of a service. The specific service the request relates to is uniquely identified by the combination of Service-Context-Id and Service-Identifier AVPs.

A usage example of this AVP is illustrated in Appendix A (Flow IX).

8.29. Rating-Group AVP

The Rating-Group AVP is of type Unsigned32 (AVP Code 432) and contains the identifier of a rating group. All the services subject to the same rating type are part of the same rating group. The specific rating group the request relates to is uniquely identified by the combination of Service-Context-Id and Rating-Group AVPs.

A usage example of this AVP is illustrated in Appendix A (Flow IX).

8.30. G-S-U-Pool-Reference AVP

The G-S-U-Pool-Reference AVP (AVP Code 457) is of type Grouped. It is used in the Credit-Control-Answer message, and associates the Granted-Service-Unit AVP within which it appears with a credit pool within the session.

The G-S-U-Pool-Identifier AVP specifies the credit pool from which credit is drawn for this unit type.

The CC-Unit-Type AVP specifies the type of units for which credit is pooled.

The Unit-Value AVP specifies the multiplier, which converts between service units of type CC-Unit-Type and abstract service units within the credit pool (and thus to service units of any other service or rating group associated with the same pool).

The G-S-U-Pool-Reference AVP is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
G-S-U-Pool-Reference ::= < AVP Header: 457 >
                        { G-S-U-Pool-Identifier }
                        { CC-Unit-Type }
                        { Unit-Value }
```

8.31. G-S-U-Pool-Identifier AVP

The G-S-U-Pool-Identifier AVP (AVP Code 453) is of type Unsigned32 and identifies a credit pool within the session.

8.32. CC-Unit-Type AVP

The CC-Unit-Type AVP (AVP Code 454) is of type Enumerated and specifies the type of units considered to be pooled into a credit pool.

The following values are defined for the CC-Unit-Type AVP:

| | |
|------------------------|---|
| TIME | 0 |
| MONEY | 1 |
| TOTAL-OCTETS | 2 |
| INPUT-OCTETS | 3 |
| OUTPUT-OCTETS | 4 |
| SERVICE-SPECIFIC-UNITS | 5 |

8.33. Validity-Time AVP

The Validity-Time AVP is of type Unsigned32 (AVP Code 448). It is sent from the credit-control server to the credit-control client. The AVP contains the validity time of the granted service units. The measurement of the Validity-Time is started upon receipt of the Credit-Control-Answer Message containing this AVP. If the granted service units have not been consumed within the validity time specified in this AVP, the credit-control client MUST send a Credit-Control-Request message to the server, with CC-Request-Type set to UPDATE_REQUEST. The value field of the Validity-Time AVP is given in seconds.

The Validity-Time AVP is also used for the graceful service termination (see section 5.6) to indicate to the credit-control client how long the subscriber is allowed to use network resources after the specified action (i.e., REDIRECT or RESTRICT_ACCESS) started. When the Validity-Time elapses, a new intermediate interrogation is sent to the server.

8.34. Final-Unit-Indication AVP

The Final-Unit-Indication AVP (AVP Code 430) is of type Grouped and indicates that the Granted-Service-Unit AVP in the Credit-Control-Answer, or in the AA answer, contains the final units for the service. After these units have expired, the Diameter credit-control client is responsible for executing the action indicated in the Final-Unit-Action AVP (see section 5.6).

If more than one unit type is received in the Credit-Control-Answer, the unit type that first expired SHOULD cause the credit-control client to execute the specified action.

In the first interrogation, the Final-Unit-Indication AVP with Final-Unit-Action REDIRECT or RESTRICT_ACCESS can also be present with no Granted-Service-Unit AVP in the Credit-Control-Answer or in the AA answer. This indicates to the Diameter credit-control client to execute the specified action immediately. If the home service provider policy is to terminate the service, naturally, the server SHOULD return the appropriate transient failure (see section 9.1) in order to implement the policy-defined action.

The Final-Unit-Action AVP defines the behavior of the service element when the user's account cannot cover the cost of the service and MUST always be present if the Final-Unit-Indication AVP is included in a command.

If the Final-Unit-Action AVP is set to TERMINATE, no other AVPs MUST be present.

If the Final-Unit-Action AVP is set to REDIRECT at least the Redirect-Server AVP MUST be present. The Restriction-Filter-Rule AVP or the Filter-Id AVP MAY be present in the Credit-Control-Answer message if the user is also allowed to access other services that are not accessible through the address given in the Redirect-Server AVP.

If the Final-Unit-Action AVP is set to RESTRICT_ACCESS, either the Restriction-Filter-Rule AVP or the Filter-Id AVP SHOULD be present.

The Filter-Id AVP is defined in [NASREQ]. The Filter-Id AVP can be used to reference an IP filter list installed in the access device by means other than the Diameter credit-control application, e.g., locally configured or configured by another entity.

The Final-Unit-Indication AVP is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Final-Unit-Indication ::= < AVP Header: 430 >
                        { Final-Unit-Action }
                        * [ Restriction-Filter-Rule ]
                        * [ Filter-Id ]
                        [ Redirect-Server ]
```

8.35. Final-Unit-Action AVP

The Final-Unit-Action AVP (AVP Code 449) is of type Enumerated and indicates to the credit-control client the action to be taken when the user's account cannot cover the service cost.

The Final-Unit-Action can be one of the following:

- TERMINATE 0
The credit-control client MUST terminate the service session. This is the default handling, applicable whenever the credit-control client receives an unsupported Final-Unit-Action value, and it MUST be supported by all the Diameter credit-control client implementations conforming to this specification.
- REDIRECT 1
The service element MUST redirect the user to the address specified in the Redirect-Server-Address AVP. The redirect action is defined in section 5.6.2.
- RESTRICT_ACCESS 2
The access device MUST restrict the user access according to the IP packet filters defined in the Restriction-Filter-Rule AVP or according to the IP packet filters identified by the Filter-Id AVP. All the packets not matching the filters MUST be dropped (see section 5.6.3).

8.36. Restriction-Filter-Rule AVP

The Restriction-Filter-Rule AVP (AVP Code 438) is of type IPFilterRule and provides filter rules corresponding to services that are to remain accessible even if there are no more service units granted. The access device has to configure the specified filter

rules for the subscriber and MUST drop all the packets not matching these filters. Zero, one, or more such AVPs MAY be present in a Credit-Control-Answer message or in an AA answer message.

8.37. Redirect-Server AVP

The Redirect-Server AVP (AVP Code 434) is of type Grouped and contains the address information of the redirect server (e.g., HTTP redirect server, SIP Server) with which the end user is to be connected when the account cannot cover the service cost. It MUST be present when the Final-Unit-Action AVP is set to REDIRECT.

It is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Redirect-Server ::= < AVP Header: 434 >
                  { Redirect-Address-Type }
                  { Redirect-Server-Address }
```

8.38. Redirect-Address-Type AVP

The Redirect-Address-Type AVP (AVP Code 433) is of type Enumerated and defines the address type of the address given in the Redirect-Server-Address AVP.

The address type can be one of the following:

- | | |
|---|---|
| IPv4 Address | 0 |
| The address type is in the form of "dotted-decimal" IPv4 address, as defined in [IPv4]. | |
| IPv6 Address | 1 |
| The address type is in the form of IPv6 address, as defined in [IPv6Addr]. The address is a text representation of the address in either the preferred or alternate text form [IPv6Addr]. Conformant implementations MUST support the preferred form and SHOULD support the alternate text form for IPv6 addresses. | |
| URL | 2 |
| The address type is in the form of Uniform Resource Locator, as defined in [URL]. | |
| SIP URI | 3 |
| The address type is in the form of SIP Uniform Resource Identifier, as defined in [SIP]. | |

8.39. Redirect-Server-Address AVP

The Redirect-Server-Address AVP (AVP Code 435) is of type UTF8String and defines the address of the redirect server (e.g., HTTP redirect server, SIP Server) with which the end user is to be connected when the account cannot cover the service cost.

8.40. Multiple-Services-Indicator AVP

The Multiple-Services-Indicator AVP (AVP Code 455) is of type Enumerated and indicates whether the Diameter credit-control client is capable of handling multiple services independently within a (sub-) session. The absence of this AVP means that independent credit-control of multiple services is not supported.

A server not implementing the independent credit-control of multiple services MUST treat the Multiple-Services-Indicator AVP as an invalid AVP.

The following values are defined for the Multiple-Services-Indicator AVP:

MULTIPLE_SERVICES_NOT_SUPPORTED 0

Client does not support independent credit-control of multiple services within a (sub-)session.

MULTIPLE_SERVICES_SUPPORTED 1

Client supports independent credit-control of multiple services within a (sub-)session.

8.41. Requested-Action AVP

The Requested-Action AVP (AVP Code 436) is of type Enumerated and contains the requested action being sent by Credit-Control-Request command where the CC-Request-Type is set to EVENT_REQUEST. The following values are defined for the Requested-Action AVP:

DIRECT_DEBITING 0

This indicates a request to decrease the end user's account according to information specified in the Requested-Service-Unit AVP and/or Service-Identifier AVP (additional rating information may be included in service-specific AVPs or in the Service-Parameter-Info AVP). The Granted-Service-Unit AVP in the Credit-Control-Answer command contains the debited units.

REFUND_ACCOUNT

1

This indicates a request to increase the end user's account according to information specified in the Requested-Service-Unit AVP and/or Service-Identifier AVP (additional rating information may be included in service-specific AVPs or in the Service-Parameter-Info AVP). The Granted-Service-Unit AVP in the Credit-Control-Answer command contains the refunded units.

CHECK_BALANCE

2

This indicates a balance check request. In this case, the checking of the account balance is done without any credit reservation from the account. The Check-Balance-Result AVP in the Credit-Control-Answer command contains the result of the balance check.

PRICE_ENQUIRY

3

This indicates a price enquiry request. In this case, neither checking of the account balance nor reservation from the account will be done; only the price of the service will be returned in the Cost-Information AVP in the Credit-Control-Answer Command.

8.42. Service-Context-Id AVP

The Service-Context-Id AVP is of type UTF8String (AVP Code 461) and contains a unique identifier of the Diameter credit-control service specific document that applies to the request (as defined in section 4.1.2). This is an identifier allocated by the service provider, by the service element manufacturer, or by a standardization body, and MUST uniquely identify a given Diameter credit-control service specific document. The format of the Service-Context-Id is:

"service-context" "@" "domain"

service-context = Token

The Token is an arbitrary string of characters and digits.

'domain' represents the entity that allocated the Service-Context-Id. It can be ietf.org, 3gpp.org, etc., if the identifier is allocated by a standardization body, or it can be the FQDN of the service provider (e.g., provider.example.com) or of the vendor (e.g., vendor.example.com) if the identifier is allocated by a private entity.

This AVP SHOULD be placed as close to the Diameter header as possible.

Service-specific documents that are for private use only (i.e., to one provider's own use, where no interoperability is deemed useful) may define private identifiers without need of coordination. However, when interoperability is wanted, coordination of the identifiers via, for example, publication of an informational RFC is RECOMMENDED in order to make Service-Context-Id globally available.

8.43. Service-Parameter-Info AVP

The Service-Parameter-Info AVP (AVP Code 440) is of type Grouped and contains service-specific information used for price calculation or rating. The Service-Parameter-Type AVP defines the service parameter type, and the Service-Parameter-Value AVP contains the parameter value. The actual contents of these AVPs are not within the scope of this document and SHOULD be defined in another Diameter application, in standards written by other standardization bodies, or in service-specific documentation.

In the case of an unknown service request (e.g., unknown Service-Parameter-Type), the corresponding answer message MUST contain the error code `DIAMETER_RATING_FAILED`. A Credit-Control-Answer message with this error MUST contain one or more Failed-AVP AVPs containing the Service-Parameter-Info AVPs that caused the failure.

It is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Service-Parameter-Info ::= < AVP Header: 440 >
                        { Service-Parameter-Type }
                        { Service-Parameter-Value }
```

8.44. Service-Parameter-Type AVP

The Service-Parameter-Type AVP is of type Unsigned32 (AVP Code 441) and defines the type of the service event specific parameter (e.g., it can be the end-user location or service name). The different parameters and their types are service specific, and the meanings of these parameters are not defined in this document. Whoever allocates the Service-Context-Id (i.e., unique identifier of a service-specific document) is also responsible for assigning Service-Parameter-Type values for the service and ensuring their uniqueness within the given service. The Service-Parameter-Value AVP contains the value associated with the service parameter type.

8.45. Service-Parameter-Value AVP

The Service-Parameter-Value AVP is of type OctetString (AVP Code 442) and contains the value of the service parameter type.

8.46. Subscription-Id AVP

The Subscription-Id AVP (AVP Code 443) is used to identify the end user's subscription and is of type Grouped. The Subscription-Id AVP includes a Subscription-Id-Data AVP that holds the identifier and a Subscription-Id-Type AVP that defines the identifier type.

It is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
Subscription-Id ::= < AVP Header: 443 >
                    { Subscription-Id-Type }
                    { Subscription-Id-Data }
```

8.47. Subscription-Id-Type AVP

The Subscription-Id-Type AVP (AVP Code 450) is of type Enumerated, and it is used to determine which type of identifier is carried by the Subscription-Id AVP.

This specification defines the following subscription identifiers. However, new Subscription-Id-Type values can be assigned by an IANA designated expert, as defined in section 12. A server MUST implement all the Subscription-Id-Types required to perform credit authorization for the services it supports, including possible future values. Unknown or unsupported Subscription-Id-Types MUST be treated according to the 'M' flag rule, as defined in [DIAMBASE].

| | |
|--|---|
| END_USER_E164 | 0 |
| The identifier is in international E.164 format (e.g., MSISDN), according to the ITU-T E.164 numbering plan defined in [E164] and [CE164]. | |

END_USER_IMSI 1

The identifier is in international IMSI format, according to the ITU-T E.212 numbering plan as defined in [E212] and [CE212].

END_USER_SIP_URI 2
The identifier is in the form of a SIP URI, as defined in [SIP].

END_USER_NAI 3
The identifier is in the form of a Network Access Identifier, as defined in [NAI].

END_USER_PRIVATE

4

The Identifier is a credit-control server private identifier.

8.48. Subscription-Id-Data AVP

The Subscription-Id-Data AVP (AVP Code 444) is used to identify the end user and is of type UTF8String. The Subscription-Id-Type AVP defines which type of identifier is used.

8.49. User-Equipment-Info AVP

The User-Equipment-Info AVP (AVP Code 458) is of type Grouped and allows the credit-control client to indicate the identity and capability of the terminal the subscriber is using for the connection to network.

It is defined as follows (per the grouped-avp-def of RFC 3588 [DIAMBASE]):

```
User-Equipment-Info ::= < AVP Header: 458 >
                        { User-Equipment-Info-Type }
                        { User-Equipment-Info-Value }
```

8.50. User-Equipment-Info-Type AVP

The User-Equipment-Info-Type AVP is of type Enumerated (AVP Code 459) and defines the type of user equipment information contained in the User-Equipment-Info-Value AVP.

This specification defines the following user equipment types. However, new User-Equipment-Info-Type values can be assigned by an IANA designated expert, as defined in section 12.

IMEISV

0

The identifier contains the International Mobile Equipment Identifier and Software Version in the international IMEISV format according to 3GPP TS 23.003 [3GPPIMEI].

MAC

1

The 48-bit MAC address is formatted as described in [RAD802.1X].

EUI64

2

The 64-bit identifier used to identify hardware instance of the product, as defined in [EUI64].

MODIFIED_EUI64

3

There are a number of types of terminals that have identifiers other than IMEI, IEEE 802 MACs, or EUI-64. These identifiers can be converted to modified EUI-64 format as described in [IPv6Addr] or by using some other methods referred to in the service-specific documentation.

8.51. User-Equipment-Info-Value AVP

The User-Equipment-Info-Value AVP (AVP Code 460) is of type OctetString. The User-Equipment-Info-Type AVP defines which type of identifier is used.

9. Result Code AVP Values

This section defines new Result-Code AVP [DIAMBASE] values that must be supported by all Diameter implementations that conform to this specification.

The Credit-Control-Answer message includes the Result-Code AVP, which may indicate that an error was present in the Credit-Control-Request message. A rejected Credit-Control-Request message SHOULD cause the user's session to be terminated.

9.1. Transient Failures

Errors that fall within the transient failures category are used to inform a peer that the request could not be satisfied at the time it was received, but that the request MAY be able to be satisfied in the future.

DIAMETER_END_USER_SERVICE_DENIED 4010

The credit-control server denies the service request due to service restrictions. If the CCR contained used-service-units, they are deducted, if possible.

DIAMETER_CREDIT_CONTROL_NOT_APPLICABLE 4011

The credit-control server determines that the service can be granted to the end user but that no further credit-control is needed for the service (e.g., service is free of charge).

DIAMETER_CREDIT_LIMIT_REACHED 4012

The credit-control server denies the service request because the end user's account could not cover the requested service. If the CCR contained used-service-units they are deducted, if possible.

9.2. Permanent Failures

Errors that fall within the permanent failure category are used to inform the peer that the request failed and should not be attempted again.

DIAMETER_USER_UNKNOWN

5030

The specified end user is unknown in the credit-control server.

DIAMETER_RATING_FAILED

5031

This error code is used to inform the credit-control client that the credit-control server cannot rate the service request due to insufficient rating input, an incorrect AVP combination, or an AVP or an AVP value that is not recognized or supported in the rating. The Failed-AVP AVP MUST be included and contain a copy of the entire AVP(s) that could not be processed successfully or an example of the missing AVP complete with the Vendor-Id if applicable. The value field of the missing AVP should be of correct minimum length and contain zeros.

10. AVP Occurrence Table

The following table presents the AVPs defined in this document and specifies in which Diameter messages they MAY or MAY NOT be present. Note that AVPs that can only be present within a Grouped AVP are not represented in this table.

The table uses the following symbols:

- | | |
|-----|---|
| 0 | The AVP MUST NOT be present in the message. |
| 0+ | Zero or more instances of the AVP MAY be present in the message. |
| 0-1 | Zero or one instance of the AVP MAY be present in the message. It is considered an error if there is more than one instance of the AVP. |
| 1 | One instance of the AVP MUST be present in the message. |
| 1+ | At least one instance of the AVP MUST be present in the message. |

10.1. Credit-Control AVP Table

The table in this section is used to represent which credit-control applications specific AVPs defined in this document are to be present in the credit-control messages.

| Attribute Name | Command Code | |
|----------------------------------|--------------|-----|
| | CCR | CCA |
| Acct-Multi-Session-Id | 0-1 | 0-1 |
| Auth-Application-Id | 1 | 1 |
| CC-Correlation-Id | 0-1 | 0 |
| CC-Session-Failover | 0 | 0-1 |
| CC-Request-Number | 1 | 1 |
| CC-Request-Type | 1 | 1 |
| CC-Sub-Session-Id | 0-1 | 0-1 |
| Check-Balance-Result | 0 | 0-1 |
| Cost-Information | 0 | 0-1 |
| Credit-Control-Failure-Handling | 0 | 0-1 |
| Destination-Host | 0-1 | 0 |
| Destination-Realm | 1 | 0 |
| Direct-Debiting-Failure-Handling | 0 | 0-1 |
| Event-Timestamp | 0-1 | 0-1 |
| Failed-AVP | 0 | 0+ |
| Final-Unit-Indication | 0 | 0-1 |
| Granted-Service-Unit | 0 | 0-1 |
| Multiple-Services-Credit-Control | 0+ | 0+ |
| Multiple-Services-Indicator | 0-1 | 0 |
| Origin-Host | 1 | 1 |
| Origin-Realm | 1 | 1 |
| Origin-State-Id | 0-1 | 0-1 |
| Proxy-Info | 0+ | 0+ |
| Redirect-Host | 0 | 0+ |
| Redirect-Host-Usage | 0 | 0-1 |
| Redirect-Max-Cache-Time | 0 | 0-1 |
| Requested-Action | 0-1 | 0 |
| Requested-Service-Unit | 0-1 | 0 |
| Route-Record | 0+ | 0+ |
| Result-Code | 0 | 1 |
| Service-Context-Id | 1 | 0 |
| Service-Identifier | 0-1 | 0 |
| Service-Parameter-Info | 0+ | 0 |

| | | | |
|---------------------|-------|-------|-------|
| Session-Id | 1 | 1 | |
| Subscription-Id | 0+ | 0 | |
| Termination-Cause | 0-1 | 0 | |
| User-Equipment-Info | 0-1 | 0 | |
| Used-Service-Unit | 0+ | 0 | |
| User-Name | 0-1 | 0-1 | |
| Validity-Time | 0 | 0-1 | |
| ----- | ----- | ----- | ----- |

10.2. Re-Auth-Request/Answer AVP Table

This section defines AVPs that are specific to the Diameter credit-control application and that MAY be included in the Diameter Re-Auth-Request/Answer (RAR/RAA) message [DIAMBASE].

Re-Auth-Request/Answer command MAY include the following additional AVPs:

| Attribute Name | Command Code | |
|-----------------------|--------------|-----|
| | RAR | RAA |
| CC-Sub-Session-Id | 0-1 | 0-1 |
| G-S-U-Pool-Identifier | 0-1 | 0-1 |
| Service-Identifier | 0-1 | 0-1 |
| Rating-Group | 0-1 | 0-1 |

11. RADIUS/Diameter Credit-Control Interworking Model

This section defines the basic principles for the Diameter credit-control/RADIUS prepaid inter-working model; that is, a message translation between a RADIUS based prepaid solution and a Diameter credit-control application. A complete description of the protocol translations between RADIUS and the Diameter credit-control application is beyond the scope of this specification and SHOULD be addressed in another appropriate document, such as the RADIUS prepaid specification.

The Diameter credit-control architecture may have a Translation Agent capable of translation between RADIUS prepaid and Diameter credit-control protocols. An AAA server (usually the home AAA server) may act as a Translation Agent and as a Diameter credit-control client for service elements that use credit-control mechanisms other than Diameter credit control for instance, RADIUS prepaid. In this case, the home AAA server contacts the Diameter credit-control server as part of the authorization process. The interworking architecture is

illustrated in Figure 7, and interworking flow in Figure 8. In a roaming situation the service element (e.g., the NAS) may be located in the visited network, and a visited AAA server is usually contacted. The visited AAA server connects then to the home AAA server.

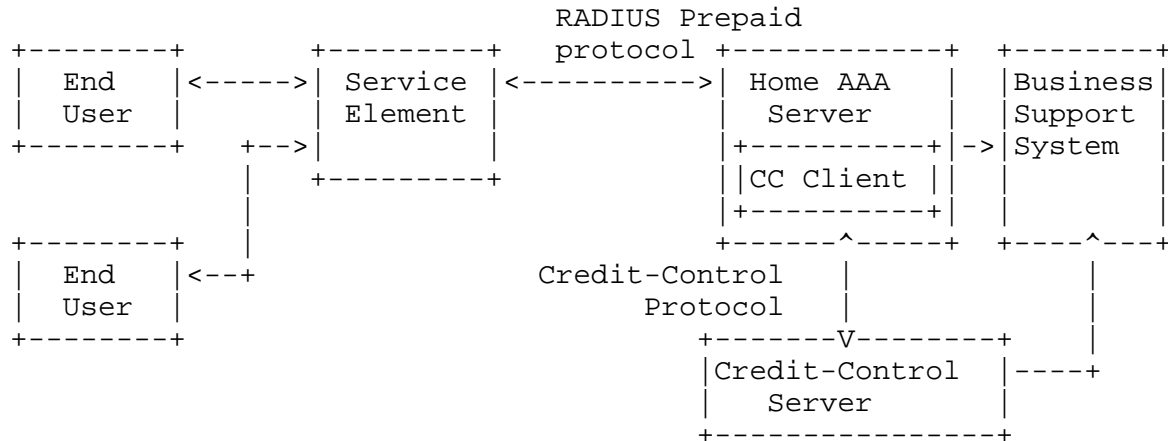


Figure 7: Credit-control architecture with service element containing translation agent, translating RADIUS prepaid to Diameter credit-control protocol

When the AAA server acting as a Translation Agent receives an initial RADIUS Access-Request message from service element (e.g., NAS access), it performs regular authentication and authorization. If the RADIUS Access-Request message indicates that the service element is capable of credit-control, and if the home AAA server finds that the subscriber is a prepaid subscriber, then a Diameter credit-control request SHOULD be sent toward the credit-control server to perform credit authorization and to establish a credit-control session. After the Diameter credit-control server checks the end user's account balance, rates the service, and reserves credit from the end user's account, the reserved quota is returned to the home AAA server in the Diameter Credit-Control-Answer. Then the home AAA server sends the reserved quota to the service element in the RADIUS Access-Accept.

At the expiry of the allocated quota, the service element sends a new RADIUS Access-Request containing the units used this far to the home AAA server. The home AAA server shall map a RADIUS Access-Request containing the reported units to the Diameter credit-control server in a Diameter Credit-Control-Request (UPDATE_REQUEST). The Diameter credit-control server debits the used units from the end user's account and allocates a new quota that is returned to the home AAA server in the Diameter Credit-Control-Answer. The quota is

transferred to the service element in the RADIUS Access-Accept. When the end user terminates the service, or when the entire quota has been used, the service element sends a RADIUS Access-Request. To debit the used units from the end user's account and to stop the credit-control session, the home AAA server sends a Diameter Credit-Control-Request (TERMINATION_REQUEST) to the credit-control server. The Diameter credit-control server acknowledges the session termination by sending a Diameter Credit-Control-Answer to the home AAA server. The RADIUS Access-Accept is sent to the NAS.

A following diagram illustrates a RADIUS prepaid - Diameter credit-control interworking sequence.

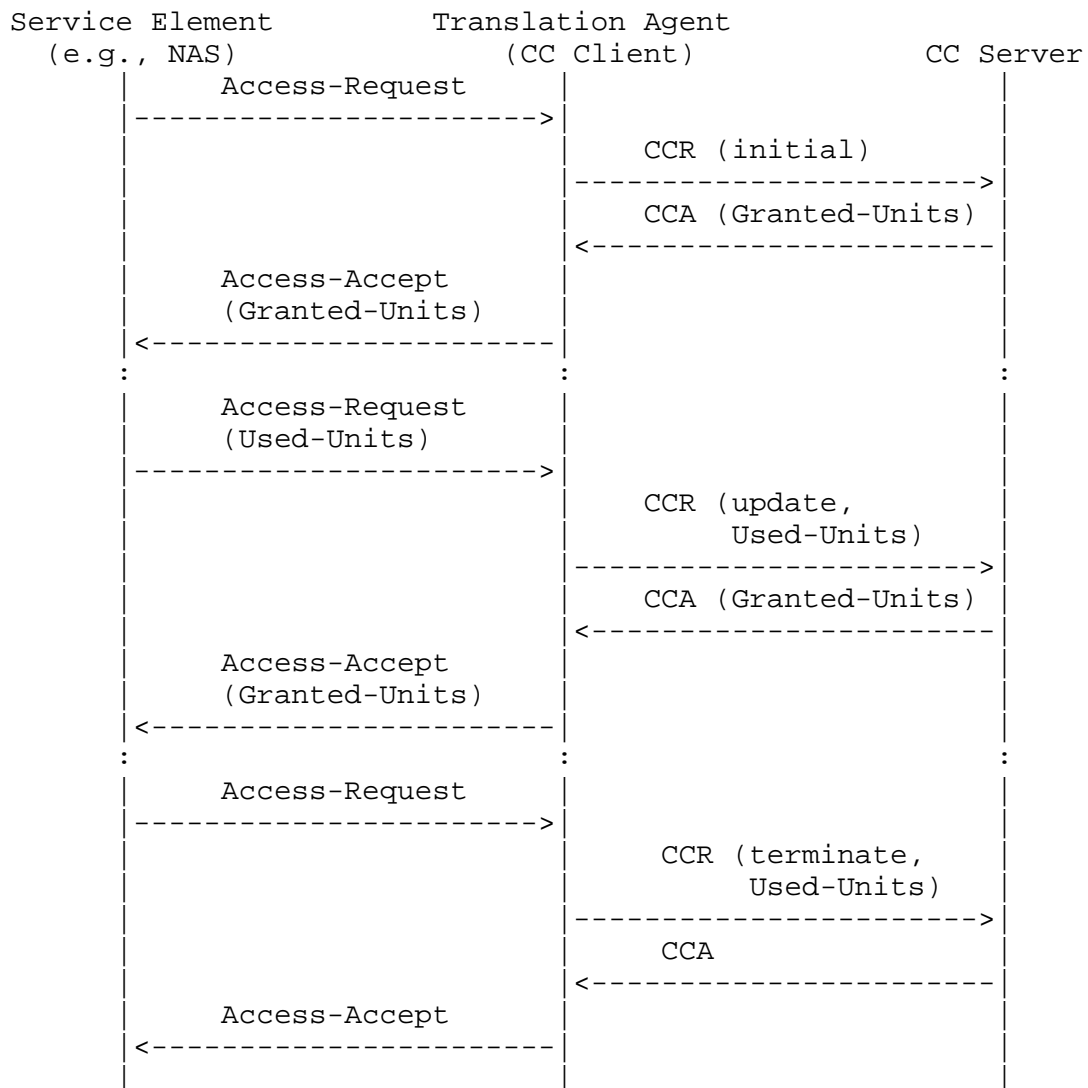


Figure 8: Message flow example with RADIUS prepaid - Diameter credit-control interworking

12. IANA Considerations

This section contains the namespaces that have either been created in this specification, or the values assigned to existing namespaces managed by IANA.

In the subsections below, when we speak about review by a Designated Expert, please note that the designated expert will be assigned by the IESG. Initially, such Expert discussions take place on the AAA WG mailing list.

12.1. Application Identifier

This specification assigns the value 4, 'Diameter Credit Control', to the Application Identifier namespace defined in [DIAMBASE]. See section 1.3 for more information.

12.2. Command Codes

This specification uses the value 272 from the Command code namespace defined in [DIAMBASE] for the Credit-Control-Request (CCR) and Credit-Control-Answer (CCA) commands.

12.3. AVP Codes

This specification assigns the values 411 - 461 from the AVP code namespace defined in [DIAMBASE]. See section 8 for the assignment of the namespace in this specification.

12.4. Result-Code AVP Values

This specification assigns the values 4010, 4011, 4012, 5030, 5031 from the Result-Code AVP value namespace defined in [DIAMBASE]. See section 9 for the assignment of the namespace in this specification.

12.5. CC-Request-Type AVP

As defined in section 8.3, the CC-Request-Type AVP includes Enumerated type values 1 - 4. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.6. CC-Session-Failover AVP

As defined in section 8.4, the CC-Failover-Supported AVP includes Enumerated type values 0 - 1. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.7. CC-Unit-Type AVP

As defined in section 8.32, the CC-Unit-Type AVP includes Enumerated type values 0 - 5. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.8. Check-Balance-Result AVP

As defined in section 8.6, the Check-Balance-Result AVP includes Enumerated type values 0 - 1. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.9. Credit-Control AVP

As defined in section 8.13, the Credit-Control AVP includes Enumerated type values 0 - 1. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.10. Credit-Control-Failure-Handling AVP

As defined in section 8.14, the Credit-Control-Failure-Handling AVP includes Enumerated type values 0 - 2. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.11. Direct-Debiting-Failure-Handling AVP

As defined in section 8.15, the Direct-Debiting-Failure-Handling AVP includes Enumerated type values 0 - 1. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.12. Final-Unit-Action AVP

As defined in section 8.35, the Final-Unit-Action AVP includes Enumerated type values 0 - 2. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.13. Multiple-Services-Indicator AVP

As defined in section 8.40, the Multiple-Services-Indicator AVP includes Enumerated type values 0 - 1. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.14. Redirect-Address-Type AVP

As defined in section 8.38, the Redirect-Address-Type AVP includes Enumerated type values 0 - 3. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.15. Requested-Action AVP

As defined in section 8.41, the Requested-Action AVP includes Enumerated type values 0 - 3. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.16. Subscription-Id-Type AVP

As defined in section 8.47, the Subscription-Id-Type AVP includes Enumerated type values 0 - 4. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.17. Tariff-Change-Usage AVP

As defined in section 8.27, the Tariff-Change-Usage AVP includes Enumerated type values 0 - 2. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

12.18. User-Equipment-Info-Type AVP

As defined in section 8.50, the User-Equipment-Info-Type AVP includes Enumerated type values 0 - 3. IANA has created and is maintaining a namespace for this AVP. All remaining values are available for assignment by a Designated Expert [IANA].

13. Credit-Control Application Related Parameters

Tx timer

When real-time credit-control is required, the credit-control client contacts the credit-control server before and while the service is provided to an end user. Due to the real-time nature of the application, the communication delays SHOULD be minimized; e.g., to avoid an overly long service setup time experienced by the end user. The Tx timer is introduced to control the waiting time in the client in the Pending state. When the Tx timer elapses, the credit-control client takes an action to the end user according to the value of the Credit-Control-Failure-Handling AVP

or Direct-Debiting-Failure-Handling AVP. The recommended value is 10 seconds.

Tcc timer

The Tcc timer supervises an ongoing credit-control session in the credit-control server. It is RECOMMENDED to use the Validity-Time as input to set the Tcc timer value. In case of transient failures in the network, the Diameter credit-control server might change to Idle state. To avoid this, the Tcc timer MAY be set so that Tcc equals to 2 x Validity-Time.

Credit-Control-Failure-Handling and Direct-Debiting-Failure-Handling

Client implementations may offer the possibility of locally configuring these AVPs. In such a case their value and behavior is defined in section 5.7 for the Credit-Control-Failure-Handling and in section 6.5 for the Direct-Debiting-Failure-Handling.

14. Security Considerations

The Diameter base protocol [DIAMBASE] requires that each Diameter implementation use underlying security; i.e., IPsec or TLS. These mechanisms are believed to provide sufficient protection under the normal Internet threat model; that is, assuming that the authorized nodes engaging in the protocol have not been compromised, but that the attacker has complete control over the communication channels between them. This includes eavesdropping, message modification, insertion, and man-in-the-middle and replay attacks. Note also that this application includes a mechanism for application layer replay protection by means of the Session-Id from [DIAMBASE] and CC-Request-Number, which is specified in this document. The Diameter credit-control application is often used within one domain, and there may be a single hop between the peers. In these environments, the use of TLS or IPsec is sufficient. The details of TLS and IPsec related security considerations are discussed in the [DIAMBASE].

Because this application handles monetary transactions (directly or indirectly), it increases the interest for various security attacks. Therefore, all parties communicating with each other MUST be authenticated, including, for instance, TLS client-side authentication. In addition, authorization of the client SHOULD be emphasized; i.e., that the client is allowed to perform credit-control for a certain user. The specific means of authorization are outside of the scope of this specification but can be, for instance, manual configuration.

Another kind of threat is malicious modification, injection, or deletion of AVPs or complete credit-control messages. The credit-control messages contain sensitive billing related information (such as subscription Id, granted units, used units, cost information) whose malicious modification can have financial consequences. Sometimes simply delaying the credit-control messages can cause disturbances in the credit-control client or server.

Even without any modification to the messages, an adversary can invite a security threat by eavesdropping, as the transactions contain private information about the user. Also, by monitoring the credit-control messages one can collect information about the credit-control server's billing models and business relationships.

When third-party relays or proxy are involved, the hop-by-hop security does not necessarily provide sufficient protection for Diameter user session. In some cases, it may be inappropriate to send Diameter messages, such as CCR and CCA, containing sensitive AVPs via untrusted Diameter proxy agents, as there are no assurances that third-party proxies will not modify the credit-control commands or AVP values.

14.1. Direct Connection with Redirects

A Diameter credit-control agent cannot always know whether agents between it and the end user's Diameter credit-control server are reliable. In this case, the Diameter credit-control agent doesn't have a routing entry in its Diameter Routing Table (defined in [DIAMBASE], section 2.7) for the realm of the credit-control server in the end user's home domain. The Diameter credit-control agent can have a default route configured to a local Redirect agent, and it redirects the CCR message to the redirect agent. The local Redirect agent then returns a redirect notification (Result-code 3006, DIAMETER_REDIRECT_INDICATION) to the credit-control agent, as well as Diameter credit-control server(s) information (Redirect-Host AVP) and information (Redirect-Host-Usage AVP) about how the routing entry resulting from the Redirect-Host is to be used. The Diameter credit-control agent then forwards the CCR message directly to one of the hosts identified by the CCA message from the redirect agent. If the value of the Redirect-Host-Usage AVP is unequal to zero, all following messages are sent to the host specified in the Redirect-Host AVP until the time specified by the Redirect-Max-Cache-Time AVP is expired.

There are some authorization issues even with redirects. There may be attacks toward nodes that have been properly authorized, but that abuse their authorization or have been compromised. These issues are discussed more widely in [DIAMEAP], section 8.

15. References

15.1. Normative References

- [DIAMBASE] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.
- [3GPPCHARG] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects, Service aspects; Charging and Billing, (release 5), 3GPP TS 22.115 v. 5.2.1, 2002-03.
- [SIP] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [NAI] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999.
- [E164] Recommendation E.164/I.331 (05/97): The International Public Telecommunication Numbering Plan. 1997.
- [CE164] Complement to ITU-T Recommendation E.164 (05/1997): "List of ITU-T Recommendation E.164 assigned country codes", June 2000.
- [E212] Recommendation E.212 (11/98): The international identification plan for mobile terminals and mobile users. 1998.
- [CE212] Complement to ITU-T Recommendation E.212 (11/1997): "List of mobile country or geographical area codes", February 1999.
- [IANA] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [IPv4] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [IPv6Addr] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.
- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [ISO4217] Codes for the representation of currencies and funds, International Standard ISO 4217, 2001
- [NASREQ] Calhoun, P., Zorn, G., Spence, D., and D. Mitton, "Diameter Network Access Server Application", RFC 4005, August 2005.
- [AAATrans] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, June 2003.
- [URL] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994.
- [RAD802.1X] Congdon, P., Aboba, B., Smith, A., Zorn, G., and J. Roese, "IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines", RFC 3580, September 2003.
- [EUI64] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", <http://standards.ieee.org/regauth/oui/tutorials/EUI64.html> March 1997.
- [3GPPIMEI] 3rd Generation Partnership Project; Technical Specification Group Core Network, Numbering, addressing and identification, (release 5), 3GPP TS 23.003 v. 5.8.0, 2003-12

15.2. Informative References

- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.
- [DIAMMIP] Calhoun, P., Johansson, T., Perkins, C., Hiller, T., and P. McCann, "Diameter Mobile IPv4 Application", RFC 4004, August 2005.
- [DIAMEAP] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", Work in Progress.
- [RFC3725] Rosenberg, J., Peterson, J., Schulzrinne, H., and G. Camarillo, "Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)", BCP 85, RFC 3725, April 2004.

16. Acknowledgements

The authors would like to thank Bernard Aboba, Jari Arkko, Robert Ekblad, Pasi Eronen, Benny Gustafsson, Robert Karlsson, Avi Lior, Paco Marin, Jussi Maki, Jeff Meyer, Anne Narhi, John Prudhoe, Christopher Richards, Juha Vallinen, and Mark Watson for their comments and suggestions.

Appendix A. Credit-Control Sequences

A.1. Flow I

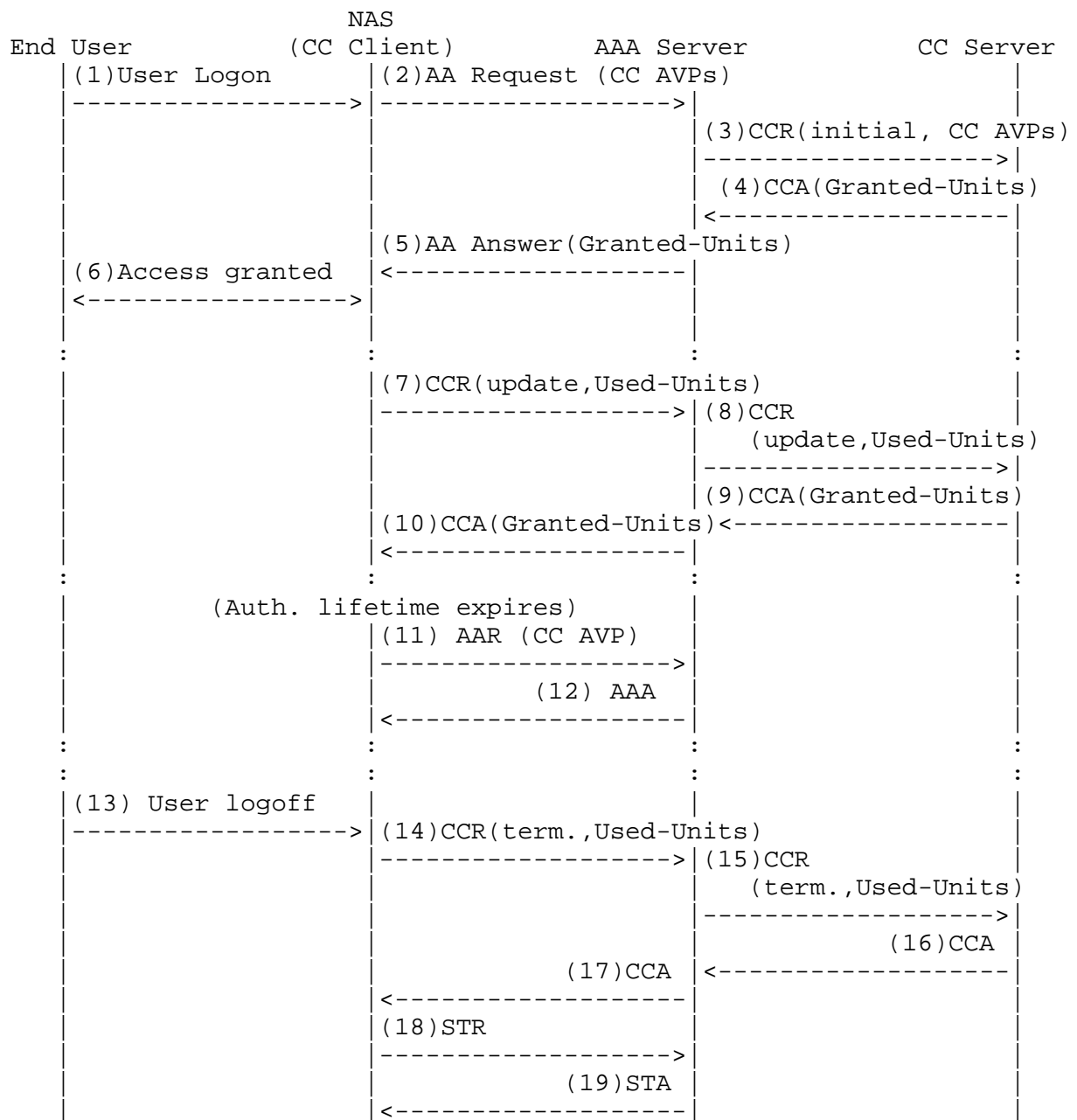


Figure A.1: Flow I

A credit-control flow for Network Access Services prepaid is shown in Figure A.1. The Diameter [NASREQ] is implemented in the Network Access Server (NAS). The focus of this flow is in the credit authorization.

The user logs on to the network (1). The Diameter NAS sends a Diameter AA-Request (AAR) to the home Diameter AAA server. The credit-control client populates the AAR with the Credit-Control AVP set to CREDIT_AUTHORIZATION, and service-specific AVPs are included, as usual [NASREQ]. The home Diameter AAA server performs service-specific Authentication and Authorization, as usual. The home Diameter AAA server determines that the user is a prepaid user and notices from the Credit-Control AVP that the NAS has credit-control capabilities. It sends a Diameter Credit-Control-Request with CC-Request-Type set to INITIAL_REQUEST to the Diameter credit-control server to perform credit authorization (3) and to establish a credit-control session. (The home Diameter AAA server may forward service-specific AVPs received from the NAS as input for the rating process.) The Diameter credit-control server checks the end user's account balance, rates the service, and reserves credit from the end user's account. The reserved quota is returned to the home Diameter AAA server in the Diameter Credit-Control-Answer (4). The home Diameter AAA server sends the reserved quota to the NAS in the Diameter AA-Answer (AAA). Upon successful AAA, the NAS starts the credit-control session and starts monitoring the granted units (5). The NAS grants access to the end user (6). At the expiry of the allocated quota, the NAS sends a Diameter Credit-Control-Request with CC-Request-Type set to UPDATE_REQUEST to the Home Diameter AAA server (7). This message contains the units used thus far. The home Diameter AAA server forwards the CCR to the Diameter credit-control server (8). The Diameter credit-control server debits the used units from the end user's account and allocates a new quota that is returned to the home Diameter AAA server in the Diameter Credit-Control-Answer (9). The message is forwarded to the NAS (10). During the ongoing credit-control session, the authorization lifetime expires, and the authorization/authentication client in the NAS performs service specific re-authorization to the home Diameter AAA server, as usual. The credit-control client populates the AAR with the Credit-Control AVP set to RE_AUTHORIZATION, indicating that the credit-control server shall not be contacted, as the credit authorization is controlled by the burning rate of the granted units (11). The home Diameter AAA server performs service-specific re-authorization as usual and returns the AA-Answer to the NAS (12). The end user logs off from the network (13). To debit the used units from the end user's account and to stop the credit-control session, the NAS sends a Diameter Credit-Control-Request with CC-Request-Type set to TERMINATION_REQUEST to the home Diameter AAA server (14). The home Diameter AAA server forwards the CCR to the credit-control

server (15). The Diameter credit-control server acknowledges the session termination by sending a Diameter Credit-Control-Answer to the home Diameter AAA server (16). The home Diameter AAA server forwards the answer to the NAS (17). STR/STA takes place between the NAS and home Diameter AAA server, as usual (18-19).

A.2. Flow II

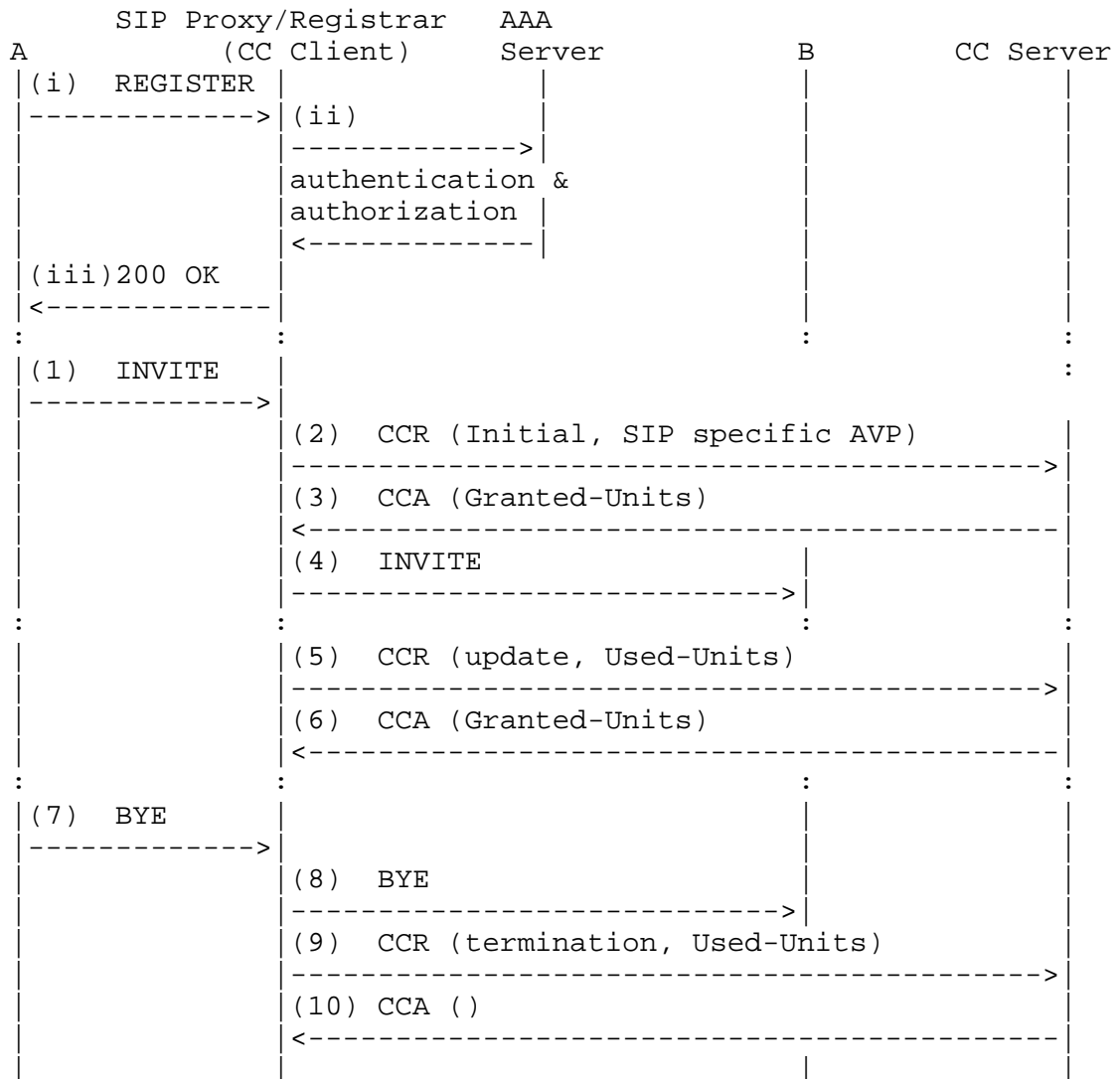


Figure A.2: Flow II

This is an example of Diameter credit-control for SIP sessions. Although the flow focuses on illustrating the usage of credit-control messages, the SIP signaling is inaccurate, and the diagram is not by any means an attempt to define a service provider's SIP network. However, for the sake of this example, some assumptions are made below.

Typically, prepaid services based, for example, on time usage for SIP session require an entity in the service provider network to intercept all the requests within the SIP dialog in order to detect events, such as session establishment and session release, that are essential to perform credit-control operations with the credit-control server. Therefore, in this example, it is assumed that the SIP Proxy adds a Record-Route header in the initial SIP INVITE to make sure that all the future requests in the created dialog traverse through it (for the definitions of 'Record-Route' and 'dialog' please refer to [SIP]). Finally, the degree of credit-control measuring of the media by the proxy depends on the business model design used in setting up the end system and proxies in the SIP network.

The end user (SIP User Agent A) sends REGISTER with credentials (i). The SIP Proxy sends a request to the home AAA server to perform Multimedia authentication and authorization by using, for instance, Diameter Multimedia application (ii). The home AAA server checks that the credentials are correct and checks the user profile. Eventually, 200 OK response (iii) is sent to the UA. Note that the Authentication and Authorization is valid for the registration validity period duration (i.e., until re-registration is performed). Several SIP sessions may be established without re-authorization.

UA A sends an INVITE (1). The SIP Proxy sends a Diameter Credit-Control-Request (INITIAL_REQUEST) to the Diameter credit-control server (2). The Credit-Control-Request contains information obtained from the SIP signaling describing the requested service (e.g., calling party, called party, Session Description Protocol attributes). The Diameter credit-control server checks the end user's account balance, rates the service, and reserves credit from the end user's account. The reserved quota is returned to the SIP Proxy in the Diameter Credit-Control-Answer (3). The SIP Proxy forwards the SIP INVITE to UA B (4). B's phone rings, and B answers. The media flows between them, and the SIP Proxy starts measuring the quota. At the expiry of the allocated quota, the SIP Proxy sends a Diameter Credit-Control-Request (UPDATE_REQUEST) to the Diameter credit-control server (5). This message contains the units used thus far. The Diameter credit-control server debits the used units from the end user's account and allocates new credit that is returned to the SIP Proxy in the Diameter Credit-Control-Answer (6). The end user terminates the service by sending a BYE (7). The SIP Proxy

forwards the BYE message to UA B (8) and sends a Diameter Credit-Control-Request (TERMINATION_REQUEST) to the credit-control server (9). The Diameter credit-control server acknowledges the session termination by sending a Diameter Credit-Control-Answer to the SIP Proxy (10).

A.3. Flow III

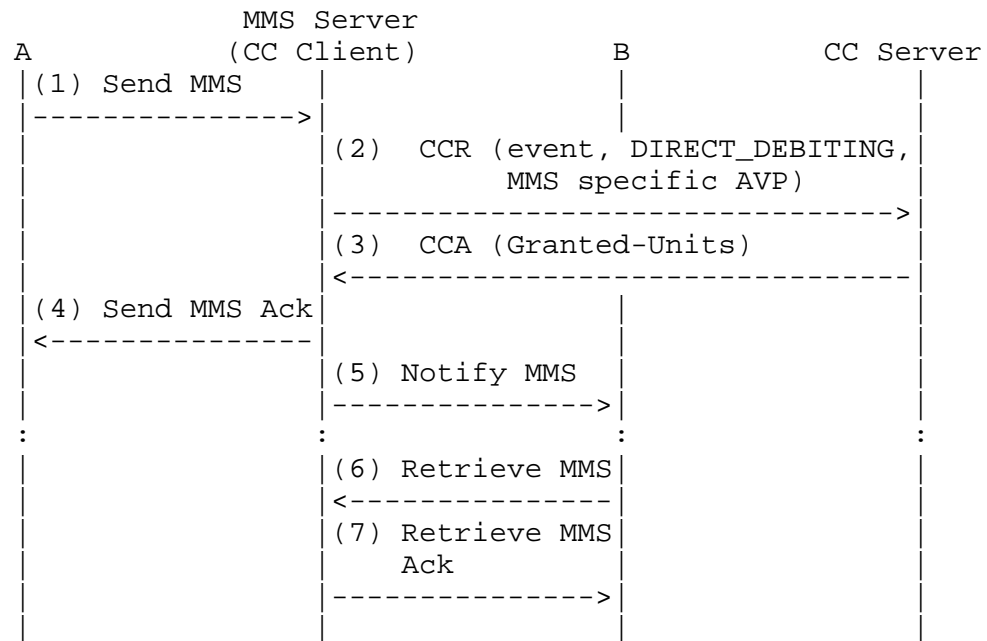


Figure A.3: Flow III

A credit-control flow for Multimedia Messaging Services is shown in Figure A.3. The sender is charged as soon as the messaging server successfully stores the message.

The end user A sends a Multimedia Message (MMS) to the MMS server (1). The MMS server stores the message and sends a Diameter Credit-Control-Request (EVENT_REQUEST with Requested-Action DIRECT_DEBITING) to the Diameter credit-control server (2). The Credit-Control-Request contains information about the MMS message (e.g., size, recipient address, image coding type). The Diameter credit-control server checks the end user's account balance, rates the service, and debits the service from the end user's account. The granted quota is returned to the MMS server in the Diameter Credit-Control-Answer (3). The MMS server acknowledges the successful reception of the MMS message (4). The MMS Server notifies the recipient about the new MMS (5), and end user B retrieves the message from the MMS message store (6), (7).

A.4. Flow IV

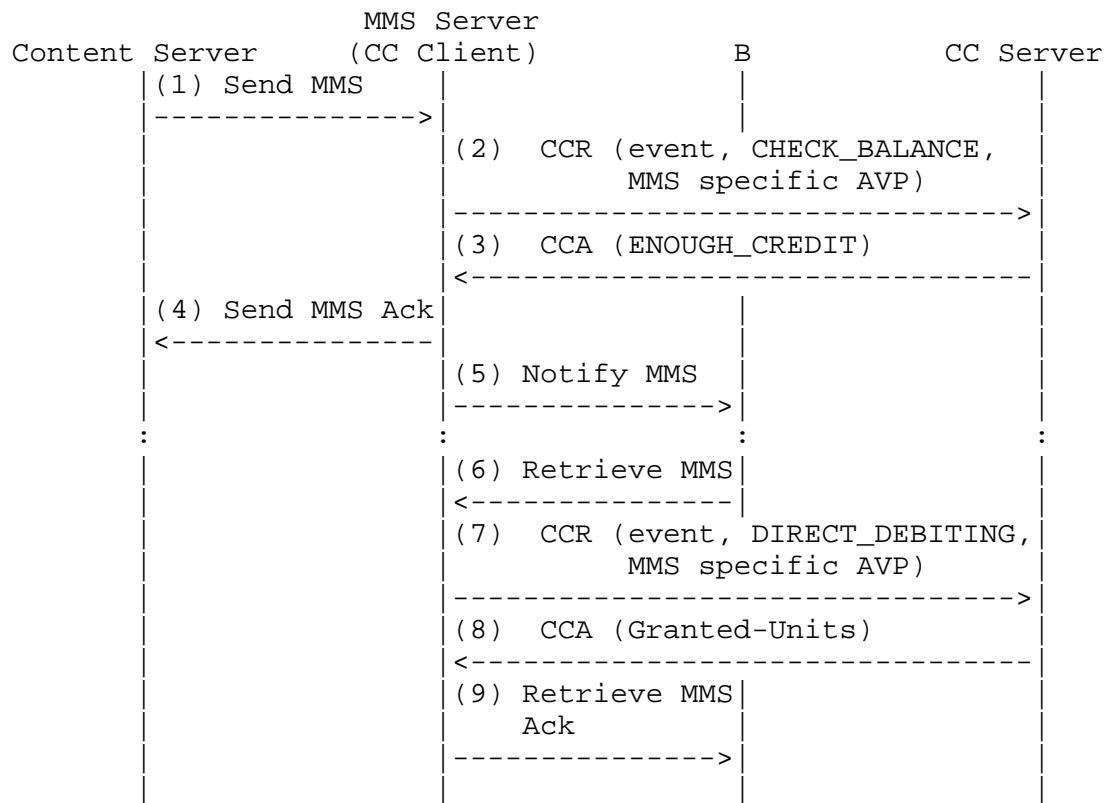


Figure A.4: Flow IV

This is an example of Diameter credit-control for direct debiting using the Multimedia Messaging Service environment. Although the flow focuses on illustrating the usage of credit-control messages, the MMS signaling is inaccurate, and the diagram is not by any means an attempt to define any service provider's MMS configuration or billing model.

A credit-control flow for Multimedia Messaging Service is shown in Figure A.4. The recipient is charged at the message delivery.

A content server sends a Multimedia Message (MMS) to the MMS server (1) that stores the message. The message recipient will be charged for the MMS message in this case. As there can be a substantially long time between the receipt of the message at the MMS server and the actual retrieval of the message, the MMS server does not establish any credit-control session to the Diameter credit-control server but performs first only a balance check (without any credit reservation) by sending a Diameter Credit-Control-Request

(EVENT_REQUEST with Requested-Action CHECK_BALANCE) to verify that end user B can cover the cost for the MMS (2). The Diameter credit-control server checks the end user's account balance and returns the answer to the MMS server in the Diameter Credit-Control-Answer (3). The MMS server acknowledges the successful reception of the MMS message (4). The MMS server notifies the recipient of the new MMS (5), and after some time end user B retrieves the message from the MMS message store (6). The MMS server sends a Diameter Credit-Control-Request (EVENT_REQUEST with Requested-Action: DIRECT_DEBITING) to the Diameter credit-control server (7). The Credit-Control-Request contains information about the MMS message (e.g., size, recipient address, coding type). The Diameter credit-control server checks the end user's account balance, rates the service, and debits the service from the end user's account. The granted quota is returned to the MMS server in the Diameter Credit-Control-Request (8). The MMS is transferred to end user B (9).

Note that the transfer of the MMS message can take an extended time and can fail, in which case a recovery action is needed. The MMS server should return the already debited units to the user's account by using the REFUND action described in section 6.4.

A.5. Flow V

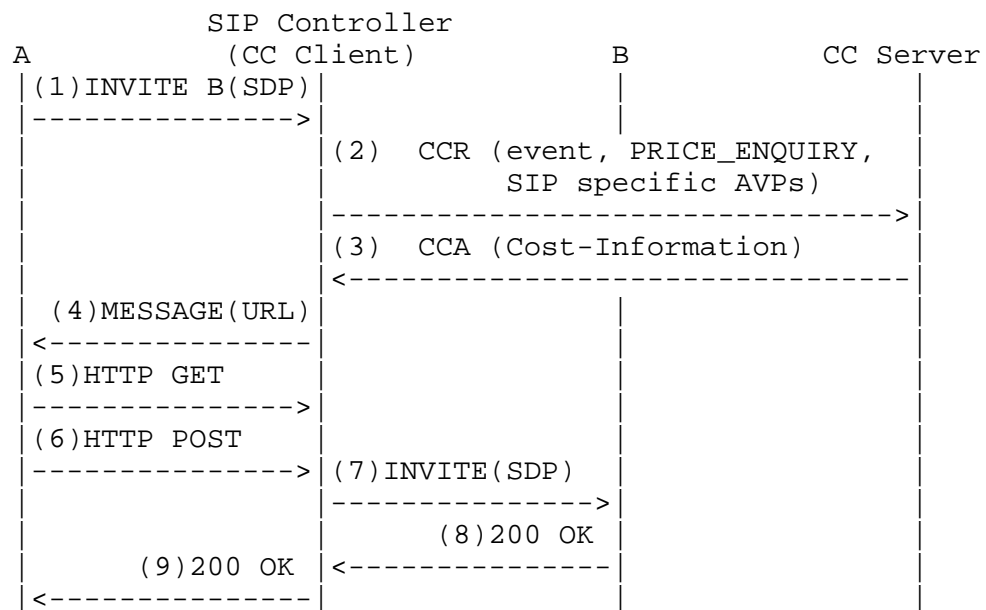


Figure A.5: Flow V

This is an example of Diameter credit-control for SIP sessions. Although the flow focuses on illustrating the usage of credit-control messages, the SIP signaling is inaccurate, and the diagram is not by any means an attempt to define a service provider's SIP network.

Figure A.5 is an example of Advice of Charge (AoC) service for SIP call. User A can be either a postpaid or prepaid subscriber using the AoC service. It is assumed that the SIP controller also has HTTP capabilities and delivers an interactive AoC web page with, for instance, the cost information, the details of the call derived from the SDP, and a button to accept/not accept the charges. (There may be many other ways to deliver AoC information; however, this flow focuses on the use of the credit-control messages.) The user has been authenticated and authorized prior to initiating the call and subscribed to AoC service.

UA A sends an INVITE with SDP to B (1). The SIP controller determines that the user is subscribed to AoC service and sends a Diameter Credit-Control-Request (EVENT_REQUEST with Requested-Action: PRICE_ENQUIRY) to the Diameter credit-control server (2). The Credit-Control-Request contains SIP specific AVPs derived from the SIP signaling, describing the requested service (e.g., calling party, called party, Session Description Protocol attributes). The Diameter credit-control server determines the cost of the service and returns the Credit-Control-Answer including the Cost-Information AVP (3). The SIP controller manufactures the AoC web page with information received in SIP signaling and with the cost information received from the credit-control server. Then it sends a SIP MESSAGE that contains a URL pointing to the AoC information web page (4). At the receipt of the SIP MESSAGE, A's UA automatically invokes the web browser that retrieves the AoC information (5). The user clicks on a proper button and accepts the charges (6). The SIP controller continues the session and sends the INVITE to the B party, which accepts the call (7,8,9).

A.6. Flow VI

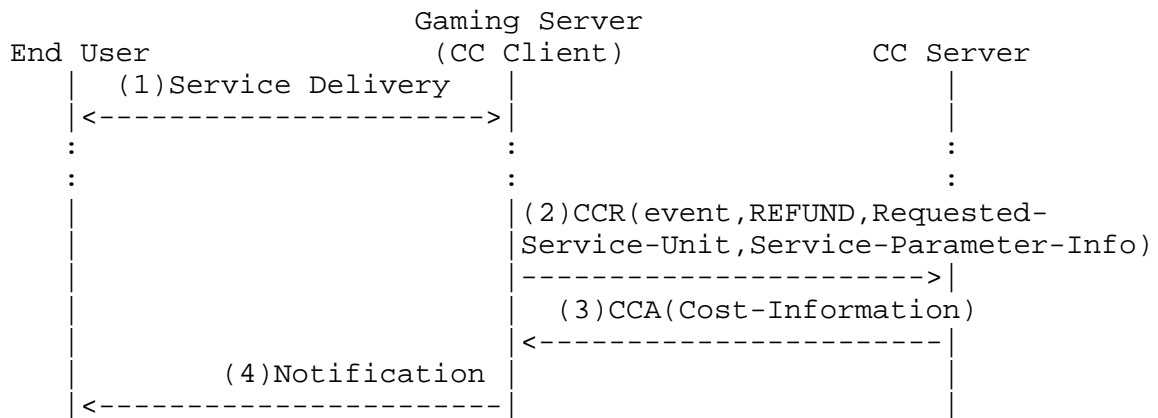


Figure A.6: Flow VI

Figure A.6 illustrates a credit-control flow for the REFUND case. It is assumed that there is a trusted relationship and secure connection between the Gaming server and the Diameter credit-control server. The end user may be a prepaid subscriber or a postpaid subscriber.

While the end user is playing the game (1), she enters a new level that entitles her to a bonus. The Gaming server sends a Diameter Credit-Control-Request (EVENT_REQUEST with Requested-Action: REFUND_ACCOUNT) to the Diameter credit-control server (2). The Credit-Control-Request contains the Requested-Service-Unit AVP with the CC-Service-Specific-Units containing the number of points the user just won. The Service-Parameter-Info AVP is also included in the request and specifies the service event to be rated (e.g., Tetris Bonus). From information received, the Diameter credit-control server determines the amount to be credited, refunds the user's account, and returns the Credit-Control-Answer, including the Cost-Information AVP (3). The Cost-Information indicates the credited amount. At the first opportunity, the Gaming server notifies the end user of the credited amount (4).

A.7. Flow VII

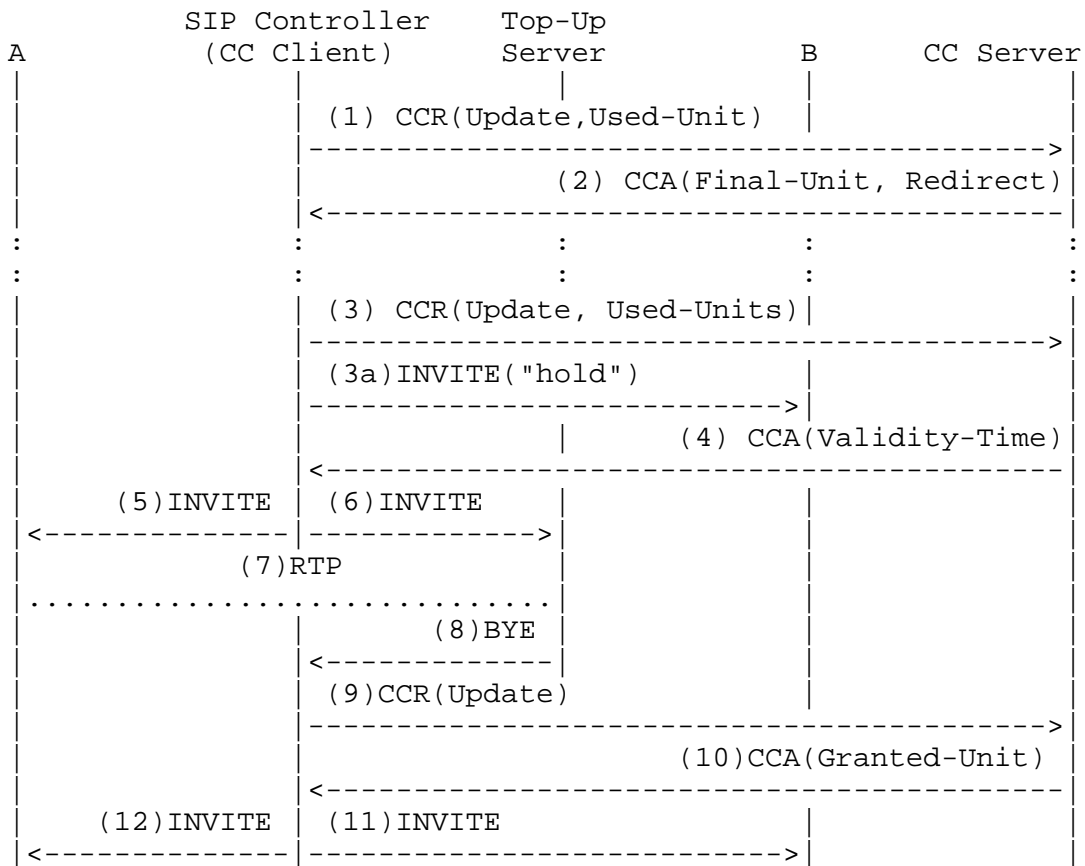


Figure A.7: Flow VII

Figure A.7 is an example of the graceful service termination for a SIP call. It is assumed that the call is set up so that the controller is in the call as a B2BUA (Back to Back User Agent) performing third-party call control (3PCC). Note that the SIP signaling is inaccurate, as the focus of this flow is in the graceful service termination and credit-control authorization. The best practice for 3PCC is defined in [RFC3725].

The call is ongoing between users A and B; user A has a prepaid subscription. At the expiry of the allocated quota, the SIP controller sends a Diameter Credit-Control-Request (UPDATE_REQUEST) to the Diameter credit-control server (1). This message contains the units used thus far. The Diameter credit-control server debits the used units from the end user's account and allocates the final quota returned to the SIP controller in the Diameter Credit-Control-Answer (2). This message contains the Final-Unit-Indication AVP with the

Final-Unit-Action set to REDIRECT, the Redirect-Address-Type set to SIP URI, and the Redirect-Server-Address set to the Top-up server name (e.g., sip:sip-topup-server@domain.com). At the expiry of the final allocated quota, the SIP controller sends a Diameter Credit-Control-Request (UPDATE_REQUEST) to the Diameter credit-control server (3) and places the called party on "hold" by sending an INVITE with the appropriate connection address in the SDP (3a). The Credit-Control-Request message contains the units used thus far. The Diameter credit-control server debits the used units from the end user's account but does not make any credit reservation. The Credit-Control-Answer message, which contains the Validity-Time to supervise the graceful service termination, is returned to the SIP controller (4). The SIP controller establishes a SIP session between the prepaid user and the Top-up server (5, 6). The Top-up server plays an announcement and prompts the user to enter a credit card number and the amount of money to be used to replenish the account (7). The Top-up server validates the credit card number and replenishes the user's account (using some means outside the scope of this specification) and releases the SIP session (8). The SIP controller can now assume that communication between the prepaid user and the Top-up server took place. It sends a spontaneous Credit-Control-Request (UPDATE_REQUEST) to the Diameter credit-control server to check whether the account has been replenished (9). The Diameter credit-control server reserves credit from the end user's account and returns the reserved quota to the SIP controller in the Credit-Control-Answer (10). At this point, the SIP controller re-connects the caller and the called party (11,12).

A.8. Flow VIII

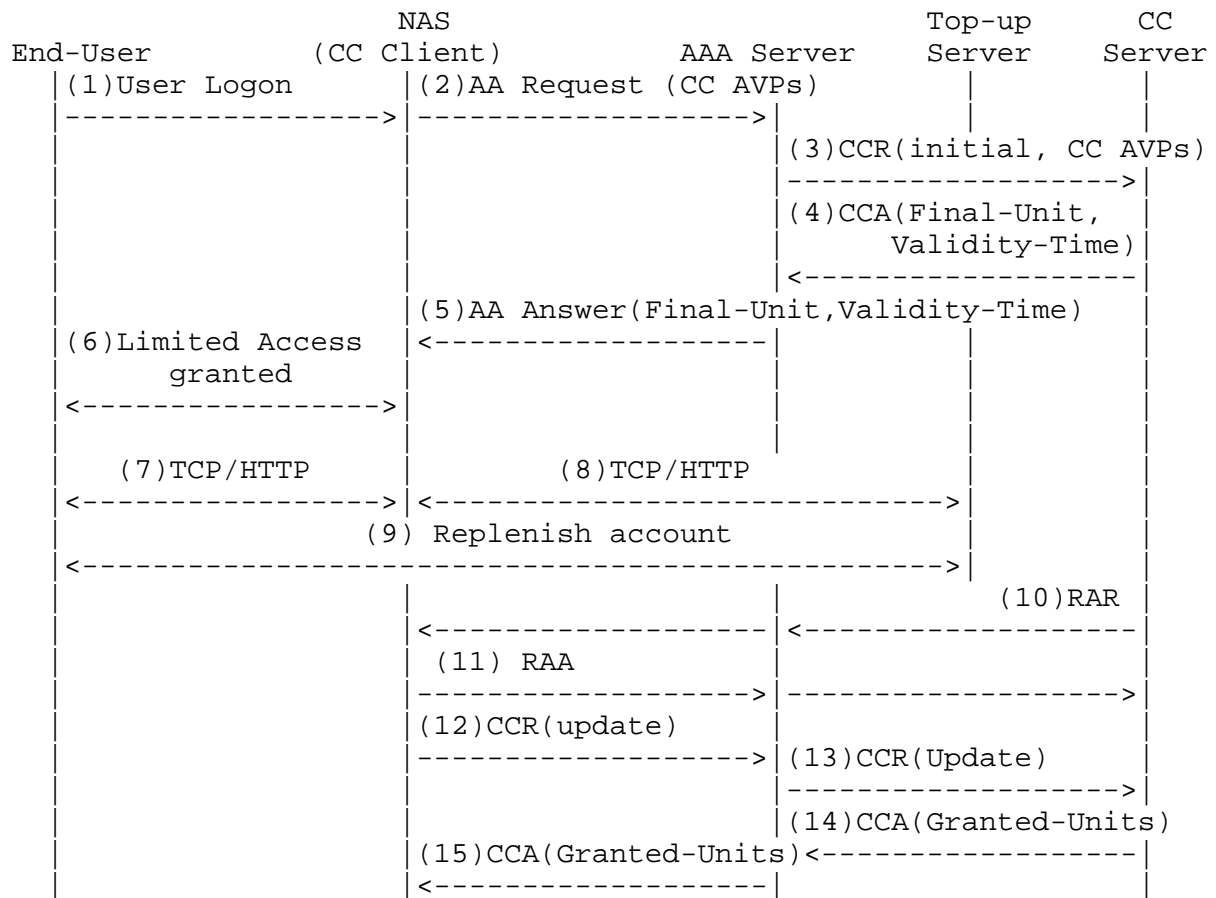


Figure A.8: Flow VIII

Figure A.8 is an example of the graceful service termination initiated when the first interrogation takes place because the user's account is empty. In this example, the credit-control server supports the server-initiated credit re-authorization. The Diameter [NASREQ] is implemented in the Network Access Server (NAS).

The user logs on to the network (1). The Diameter NAS sends a Diameter AA-Request to the home Diameter AAA server. The credit-control client populates the AAR with the Credit-Control AVP set to CREDIT_AUTHORIZATION, and service specific AVPs are included, as usual [NASREQ]. The home Diameter AAA server performs service specific Authentication and Authorization, as usual. The home Diameter AAA server determines that the user has a prepaid subscription and notices from the Credit-Control AVP that the NAS has credit-control capabilities. It sends a Diameter Credit-Control-

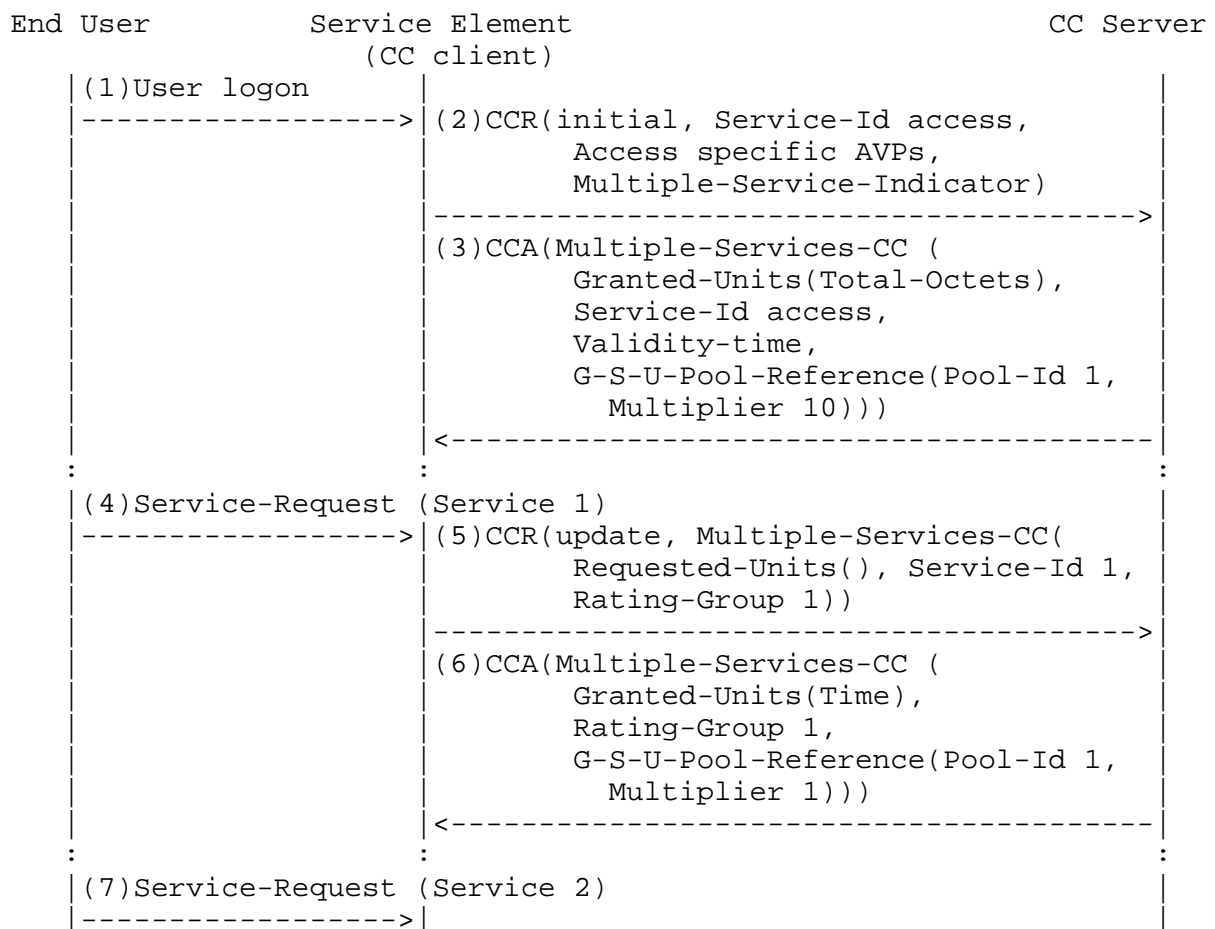
Request with CC-Request-Type set to INITIAL_REQUEST to the Diameter credit-control server to perform credit authorization (3) and to establish a credit-control session. (The home Diameter AAA server may forward service specific AVPs received from the NAS as input for the rating process.) The Diameter credit-control server checks the end user's account balance, determines that the account cannot cover the cost of the service, and initiates the graceful service termination. The Credit-Control-Answer is returned to the home Diameter AAA server (4). This message contains the Final-Unit-Indication AVP and the Validity-Time AVP set to a reasonable amount of time to give the user a chance to replenish his/her account (e.g., 10 minutes). The Final-Unit-Indication AVP includes the Final-Unit-Action set to REDIRECT, the Redirect-Address-Type set to URL, and the Redirect-Server-Address set to the HTTP Top-up server name. The home Diameter AAA server sends the received credit-control AVPs to the NAS in the Diameter AA-Answer (5). Upon successful AAA, the NAS starts the credit-control session and immediately starts the graceful service termination, as instructed by the server. The NAS grants limited access to the user (6). The HTTP client software running in the user's device opens the transport connection redirected by the NAS to the Top-up server (7,8). The user is displayed an appropriate web page on which to enter the credit card number, and the amount of money to be used to replenish the account, and with a notification message that she is granted unlimited access if the replenishment operation will be successfully executed within the next, for example, 10 minutes. The Top-up server validates the credit card number and replenishes the user's account (using some means outside the scope of this specification)(9). After successful account top-up, the credit-control server sends a Re-Auth-Request message to the NAS (10). The NAS acknowledges the request by returning the Re-Auth-Answer message (11) and initiates the credit re-authorization by sending a Credit-Control-request (UPDATE_REQUEST) to the Diameter credit-control server (12,13).

The Diameter credit-control server reserves credit from the end user's account and returns the reserved quota to the NAS via the home Diameter AAA server in the Credit-Control-Answer (14,15). The NAS removes the restriction placed by the graceful service termination and starts monitoring the granted units.

A.9. Flow IX

The Diameter credit-control application defines the Multiple-Services-Credit-Control AVP that can be used to support independent credit-control of multiple services in a single credit-control (sub-) session for service elements that have such capabilities. It is possible to request and allocate resources as a credit pool that is shared between services or rating groups.

The flow example hereafter illustrates a usage scenario where the credit-control client and server support independent credit-control of multiple services, as defined in section 5.1.2. It is assumed that Service-Identifiers, Rating-Groups, and their associated parameters (e.g., IP 5-tuple) are locally configured in the service element or provisioned by an entity other than the credit-control server.



```

:                                     :
:                                     :
: (8)Service-Request (Service 3&4)  :
: -----> (9)CCR(update, Multiple-Services-CC (
:           Requested-Units(), Service-Id 3,
:           Rating-Group 2),
:           Multiple-Services-CC (
:           Requested-Units(), Service-Id 4,
:           Rating-Group 3))
:                                     :
:                                     : ----->
: (10)CCA(Multiple-Services-CC (
:         Granted-Units(Total-Octets),
:         Service-Id 3, Rating-Group 2,
:         Validity-time,
:         G-S-U-Pool-Reference(Pool-Id 2,
:         Multiplier 2)),
:         Multiple-Services-CC (
:         Granted-Units(Total-Octets),
:         Service-Id 4, Rating-Group 3
:         Validity-Time,
:         Final-Unit-Ind.(Terminate),
:         G-S-U-Pool-Reference(Pool-Id 2,
:         Multiplier 5)))
:                                     :
:                                     : -----<
:                                     :
:                                     :
: +-----+
: | Validity time |
: | expires for   |
: | Service-Id    |
: | access        |
: +-----+
:                                     :
:                                     : ----->
: (11)CCR(update,
:         Multiple-Services-CC (
:         Requested-Unit(),
:         Used-Units(In-Octets,Out-Octets),
:         Service-Id access))
:                                     :
:                                     : ----->
: (12)CCA(Multiple-Services-CC (
:         Granted-Units(Total-Octets),
:         Service-Id access,
:         Validity-Time,
:         G-S-U-Pool-Reference(Pool-Id 1,
:         Multiplier 10)))
:                                     :
:                                     : -----<
:                                     :
:                                     :

```

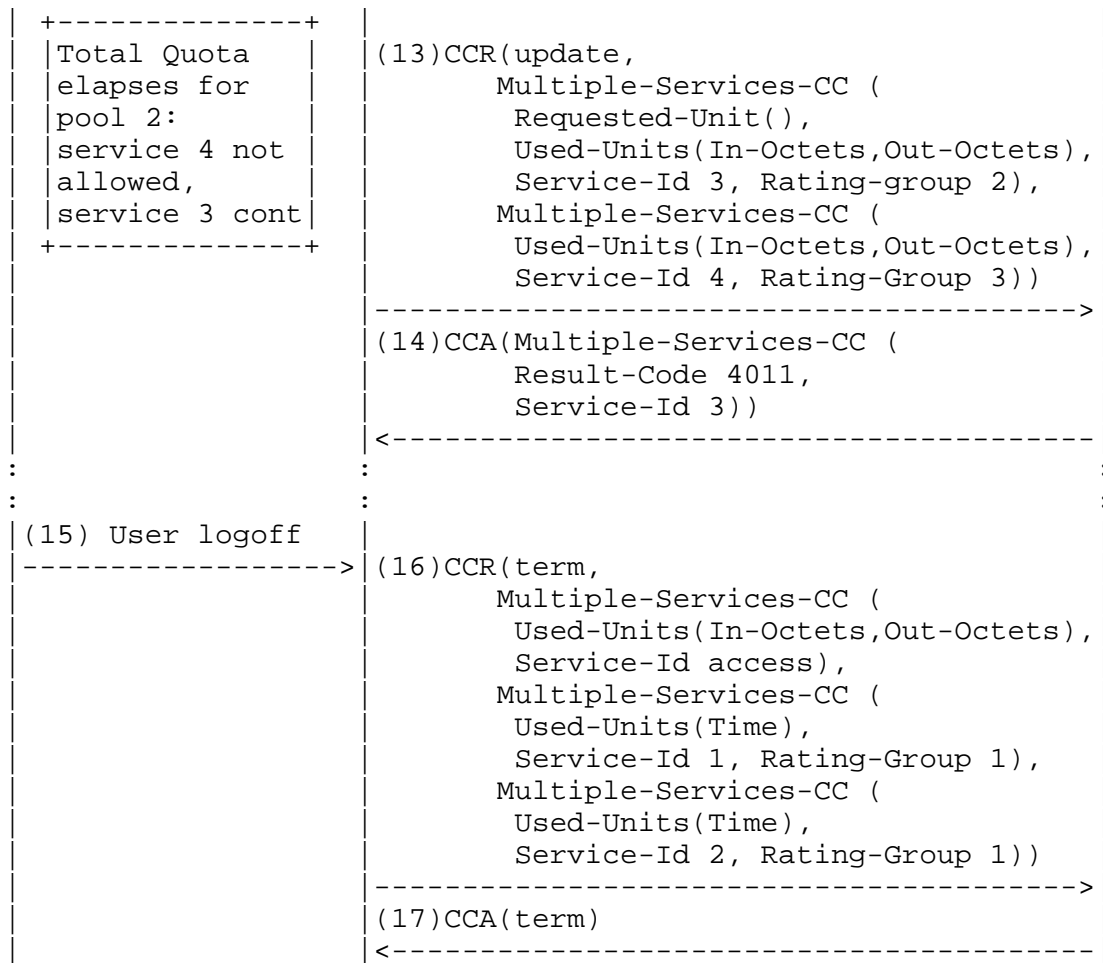


Figure A.9: Flow example independent credit-control of multiple services in a credit-control (sub-)Session

The user logs on to the network (1). The service element sends a Diameter Credit-Control-Request with CC-Request-Type set to INITIAL_REQUEST to the Diameter credit-control server to perform credit authorization for the bearer service (e.g., Internet access service) and to establish a credit-control session (2). In this message, the credit-control client indicates support for independent credit-control of multiple services within the session by including the Multiple-Service-Indicator AVP. The Diameter credit-control server checks the end user's account balance, with rating information received from the client (i.e., Service-Id and access specific AVPs), rates the request, and reserves credit from the end user's account. Suppose that the server reserves \$5 and determines that the cost is \$1/MB. It then returns to the service element a Credit-Control-Answer message that includes the Multiple-Services-Credit-Control AVP

with a quota of 5MB associated to the Service-Id (access), to a multiplier value of 10, and to the Pool-Id 1 (3).

The user uses Service 1 (4). The service element sends a Diameter Credit-Control-Request with CC-Request-Type set to UPDATE_REQUEST to the credit-control server to perform credit authorization for service 1 (5). This message includes the Multiple-Services-Credit-Control AVP to request service units for Service 1 that belong to Rating-Group 1. The Diameter credit-control server determines that Service 1 draws credit resources from the same account as the access service (i.e., pool 1). It rates the request according to Service-Id/Rating-Group and updates the existing reservation by requesting more credit. Suppose that the server reserves \$5 more (now the reservation is \$10) and determines that the cost is \$0.1/minute. The server authorizes the whole Rating-Group. It then returns to the service element a Credit-Control-Answer message that includes the Multiple-Services-Credit-Control AVP with a quota of 50min. associated to the Rating-Group 1, to a multiplier value of 1, and to the Pool-Id 1 (6). The client adjusts the total amount of resources for pool 1 according the received quota, which gives S for Pool 1 = 100.

The user uses Service 2, which belongs to the authorized Rating-Group, 1 (7). Resources are then consumed from the pool 1.

The user now requests Services 3 and 4 as well, which are not authorized (8). The service element sends a Diameter Credit-Control-Request with CC-Request-Type set to UPDATE_REQUEST to the credit-control server in order to perform credit authorization for Services 3 and 4 (9). This message includes two instances of the Multiple-Services-Credit-Control AVP to request service units for Service 3 that belong to Rating-Group 2 and for Service 4 that belong to Rating-Group 3. The Diameter credit-control server determines that Services 3 and 4 draw credit resources from another account (i.e., pool 2). It checks the end user's account balance and, according to Service-Ids/Rating-Groups information, rates the request. Then it reserves credit from pool 2.

For example, the server reserves \$5 and determines that Service 3 costs \$0.2/MB and Service 4 costs \$0.5/MB. The server authorizes only Services 3 and 4. It returns to the service element a Credit-Control-Answer message that includes two instances of the Multiple-Services-Credit-Control AVP (10). One instance grants a quota of 12.5MB associated to the Service-Id 3 to a multiplier value of 2 and to the Pool-Id 2. The other instance grants a quota of 5 MB associated to the Service-Id 4 to a multiplier value of 5 and to the Pool-Id 2.

The server also determines that pool 2 is exhausted and Service 4 is not allowed to continue after these units will be consumed. Therefore the Final-Unit-Indication AVP with action TERMINATE is associated to the Service-Id 4. The client calculates the total amount of resources that can be used for pool 2 according the received quotas and multipliers, which gives S for Pool 2 = 50.

The Validity-Time for the access service expires. The service element sends a Credit-Control-Request message to the server in order to perform credit re-authorization for Service-Id (access) (11). This message carries one instance of the Multiple-Services-Credit-Control AVP that includes the units used by this service. Suppose that the total amount of used units is 4MB. The client adjusts the total amount of resources for pool 1 accordingly, which gives S for Pool 1 = 60.

The server deducts \$4 from the user's account and updates the reservation by requesting more credit. Suppose that the server reserves \$5 more (now the reservation is \$11) and already knows the cost of the Service-Id (access), which is \$1/MB. It then returns to the service element a Credit-Control-Answer message that includes the Multiple-Services-Credit-Control AVP with a quota of 5 MB associated to the Service-Id (access), to a multiplier value of 10, and to the Pool-Id 1 (12). The client adjusts the total amount of resources for pool 1 according the received quota, which gives S for Pool 1 = 110.

Services 3 and 4 consume the total amount of pool 2 credit resources (i.e., $C1*2 + C2*5 \geq S$). The service element immediately starts the TERMINATE action concerning Service 4 and sends a Credit-Control-Request message with CC-Request-Type set to UPDATE_REQUEST to the credit-control server in order to perform credit re-authorization for Service 3 (13). This message contains two instances of the Multiple-Services-Credit-Control AVP to report the units used by Services 3 and 4. The server deducts the last \$5 from the user's account (pool 2) and returns the answer with Result-Code 4011 in the Multiple-Services-Credit-Control AVP to indicate that Service 3 can continue without credit-control (14).

The end user logs off from the network (15). To debit the used units from the end user's account and to stop the credit-control session, the service element sends a Diameter Credit-Control-Request with CC-Request-Type set to TERMINATION_REQUEST to the credit-control server (16). This message contains the units consumed by each of the used services in multiple instances of the Multiple-Services-Credit-Control AVP. The used units are associated with the relevant Service-Identifier and Rating-Group. The Diameter credit-control

server debits the used units to the user's account (Pool 1) and acknowledges the session termination by sending a Diameter Credit-Control-Answer to the service element (17).

Authors' Addresses

Harri Hakala
Oy L M Ericsson Ab
Joukahaisenkatu 1
20520 Turku
Finland

Phone: +358 2 265 3722
EMail: Harri.Hakala@ericsson.com

Leena Mattila
Oy L M Ericsson Ab
Joukahaisenkatu 1
20520 Turku
Finland

Phone: +358 2 265 3731
EMail: Leena.Mattila@ericsson.com

Juha-Pekka Koskinen
Nokia Networks
Hatanpaanvaltatie 30
33100 Tampere
Finland

Phone: +358 7180 74027
EMail: juha-pekka.koskinen@nokia.com

Marco Stura
Nokia Networks
Hiomotie 32
00380 Helsinki
Finland

Phone: +358 7180 64308
EMail: marco.stura@nokia.com

John Loughney
Nokia Research Center
Itamerenkatu 11-13
00180 Helsinki
Finland

Phone: +358 50 483 642
EMail: John.Loughney@nokia.com

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

