

Diameter Maintenance and Extensions
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2017

L. Bertz
M. Bales
Sprint
June 29, 2017

Diameter Policy Groups and Sets
draft-bertz-dime-policygroups-04

Abstract

This document defines optional Diameter attributes for efficient policy provisioning.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November

10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

1. Introduction

As Users connect to a network, policy applications often apply common policies to them. In some cases policies are grouped and applied through the use of AVPs, e.g. 3GPP Base Name. Other options include sending identifiers, usually a list of integers, associated with rules to apply a group to a single user. This compacts the over the wire representation but requires strong coordination between policy based Clients and Servers.

Application of common policy is further limited when the filters overlap. This requires partitioning policies into non-overlapping namespaces, e.g. tables in a Software Defined Networking (SDN) switch. To reduce the need to partition sets of policies some SDN technologies, e.g. OpenFlow, rely on metadata that is applied as part of the filter or metadata that is specific to the packet, e.g. OpenFlow Registers.

This document defines grouping mechanisms to allow users or groups of users to share policies or groups of policies. The mechanism also extends filters to include a metadata matching field that permits filters that overlap at the protocol level to coexist in the same policy enforcement space.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

Authorized Users An Entity that has been authorized to use a service via a Diameter Application.

Base Name An organizational structure used to define a domain for multiple Policy Groups or Membership Domains.

Determination Type The matching policy applied, e.g. ANDMASK, AND, etc, for Membership Determination.

Policy Entity A type that may be assigned to a Policy Group or Membership. This includes but is not limited to Filters [RFC7155] or Filter-Rules [RFC5777].

Membership Determination The process by which Policy Entities are selected to be applied to an authorized User.

Membership Domain A name assigned to a Membership Set.

Membership Value A binary set of values where each bit represents a specific membership pattern. This metadata is used as part of the filter or as user information when policy application occurs.

4. Concepts

Policy Groups represent a union of Policy Entities. These entities MUST be of the same type, e.g. Filters [RFC7155] or Filter-Rules [RFC5777].

When establishing groups and membership Sets an optional Base Name MAY be used. It identifies the top level grouping. Policy Entity groups MAY be directly named as well. A Policy Entity's name MUST contain zero or 1 separator character '/'. The value before the separator is a Base Name. When no Base Name is provided, i.e. no separator is present. The value of a policy entity is considered to be part of the Base Name "" (empty string) for any matching purposes. Base Name values MUST NOT contain the '/' character.

A Policy Entity can be applied to multiple, distinct sets of authorized Users. These sets can be based upon their state (paid, past due, etc.), customer type (pre-paid, post-paid, etc.) or many other factors. In such cases, a Membership Domain is used.

Membership Domains are named domains (UTF8Strings) with binary values stored in bit strings to represent where the Policy Entity is used. A Policy Entity MAY appear in multiple Membership Domains.

Membership-Value is a compact bit pattern to be used which notes when a Policy Entity or Policy Group applies to to an Authorized User.

An Authorized User's memberships are assigned by a Policy-Membership. A Policy Entity is assigned membership via a Membership-Assignment. Multiple assignments may be applied to an Authorized User and Policy Entity but they MUST have unique Membership Domain values. It is also RECOMMENDED to avoid numerous Policy-Membership assignments for

an Authorized User as it delays computation of the Policy Entities that should be applied to their service.

Memberships are matched by understanding the relationship between their values which are represented as sets of bits. These relationships are described as Match-Types and are specified as set relations, e.g. subset, superset, etc. Figure 1 shows the reference model.

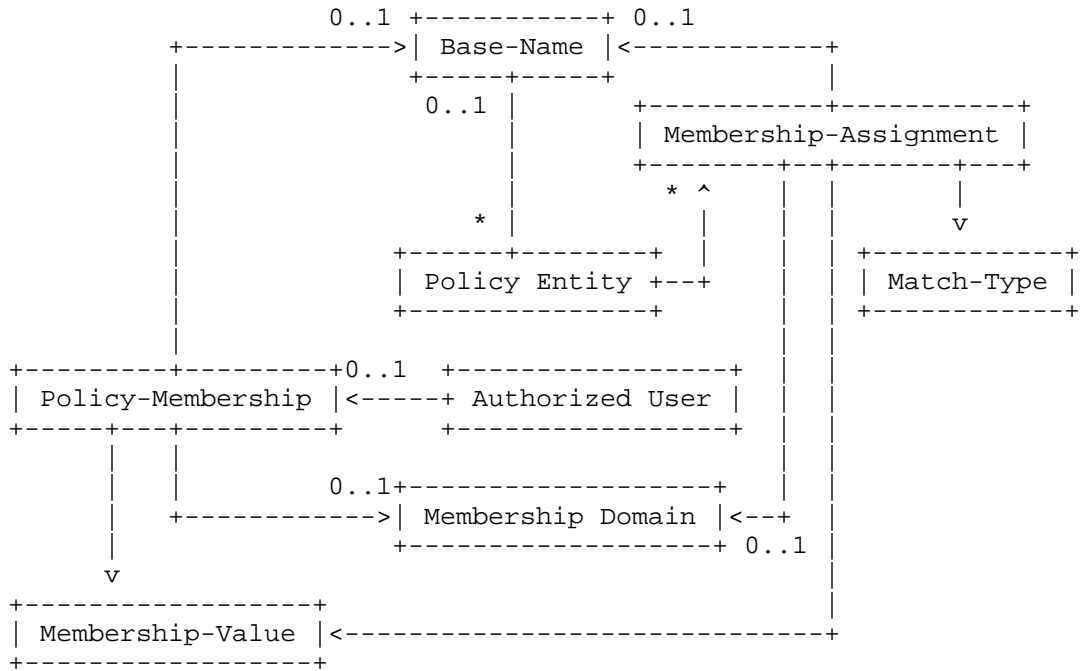


Figure 1: Reference Model

To determine if a Rule is assigned to the User the following conditions MUST be true at least one Membership-Assignments must exist where

Policy-Membership's Membership-Domain = Membership-Assignment's Membership-Domain

Policy-Membership's Membership-Value MUST satisfy the Match-Type for the Membership-Assignments' Membership-Value

5. Groups and Membership AVPs

5.1. Base-Name AVP

The Base-Name AVP (AVP Code TBD1) is of type UTF8String and defines a group of Policy Entities, e.g. Filters [RFC7155] or Filter-Rules [RFC5777].

All Policy Entities with the same Base-Name MUST be of the same AVP type.

A Base-Name MAY be assigned at the creation of the Policy Entity or in a subsequent update but MUST only be assigned once, i.e. re-assignment of the Base-Name MUST NOT be allowed.

5.2. Policy-Membership AVP

The Policy-Membership AVP (AVP Code TBD2) is of type Grouped and specifies the Membership-Value and optionally the Membership-Domain and Base-Name for an Authorized User. It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Policy-Membership ::= < AVP Header: TBD2 >
                        { Membership-Value }
                        [ Membership-Domain ]
                        [ Base-Name ]
```

Multiple Policy-Membership values MAY be assigned to an Authorized User. However, assigning multiple Policy-Memberships to an Authorized Users MAY delay policy enforcement as membership determination time is increased and SHOULD be avoided.

If multiple Policy-Memberships are assigned to an Authorized User, the Membership-Domain of each Policy-Membership value MUST be unique.

5.3. Membership-Assignment AVP

The Membership-Assignment AVP (AVP Code TBD3) is of type Grouped and specifies the Membership-Value and optionally the Membership-Domain and Base-Name for a Policy-Entity. It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Membership-Assignment ::= < AVP Header: TBD3 >
    { Membership-Value }
    { Match-Type }
    [ Membership-Domain ]
    [ Base-Name ]
```

Multiple Policy-Membership values MAY be assigned to a Policy Entity. If multiple Policy-Memberships are assigned, the Membership-Domain of each Membership-Assignment MUST be unique.

5.4. Membership-Domain AVP

The Membership-Domain AVP (AVP Code TBD4) is of type UTF8String and defines a membership set for a group of Policy Entities, e.g. Filters [RFC7155] or Filter-Rules [RFC5777], that are commonly applied to a set of Authorized Users.

5.5. Membership-Value AVP

The Membership-Value AVP (AVP Code TBD5) is of type OctetString and defines a membership of a Policy Entity or Authorized User.

Each bit of the OctetString represents a single position in the Membership-Domain set.

When two Membership-Values of different lengths are compared, the smaller Membership-Value is padded with '0' valued bits until it is the same length as the longer Membership-Value.

5.6. Match-Type AVP

The Match-Type AVP (AVP Code TBD6) is of type Enumerated and defines the type of Matching algorithm used for the Policy Entity.

When applying the Match-Type between the Membership-Value of Membership-Assignment (Policy Entity) and a Policy-Membership (Authorized User), the Membership-Domain MUST be the same, i.e. they are omitted or both MUST be present and have the same value.

Match-Types can be one of the following:

EQ 0

The Membership-Values are equal.

SUPER 1

The Membership-Assignment's Membership-Value is a superset of the Policy-Membership's Membership-Value, i.e. they may be equal.

PSUPER 2

The Membership-Assignment's Membership-Value is a proper superset of the Policy-Membership's Membership-Value.

SUB 3

The Membership-Assignment's Membership-Value is a subset of the Policy-Membership's Membership-Value, i.e. they may be equal.

PSUB 4

The Membership-Assignment's Membership-Value is a proper subset of the Policy-Membership's Membership-Value.

OVERLAP 5

The Membership-Assignment's Membership-Value has overlap with the Policy-Membership's Membership-Value. They may be equal or have some form of subset / superset relationship.

NONOVERLAP 6

The Membership-Assignment's Membership-Value has no intersection with the Policy-Membership's Membership-Value.

6. Lifecycle Considerations

Base Names are typically assigned when a Policy Entity is installed on the Diameter Client. Assignment MAY occur after installation but the impact of this is outside of the scope of this document.

Membership-Assignments MAY occur at any time in the lifecycle of the Policy Entity. However, there is no guarantee that resources exist on the Diameter Client to perform a re-evaluation of the membership of all Authorized Users. A Diameter Server MUST NOT assume that re-evaluation will occur or that an evaluation will occur immediately.

Policy-Memberships MAY change at any time in the lifecycle of the Authorized User's session. It is expected that sufficient resources exist to perform a re-evaluation of applicable Policy Entities based upon Membership testing. If this cannot be done a Diameter Application level appropriate message MUST be sent to the Diameter Server.

Generally, Base-Name assignment SHOULD occur upon creation of a Policy Entity or the authorization of a User. Membership-Assignments SHOULD occur prior to an Authorized User being created with a Policy-Membership that would apply the Policy Entity to the Authorized User's session.

7. Examples

7.1. Rule Sets

A policy administrator defines Product X with 3 separate rules sets. The administrator creates the Membership-Domain "Product X" and Membership-Values of 1, 2 and 4 representing separate rule sets. For this example each rule set consists of twenty Filter-Rules as defined in [RFC5777].

Each Rule Set is assigned a Membership-Value. Rule Set 1 is assigned a Membership-Value of 1, Rule Set 2 members is assigned the value 2 and Rule Set three members are assigned a value of 4. All Membership-Assignments have the Membership-Domain of "Product X" and a Match-Type of EQ (Equals).

The policy administrator defines three users. User 1 is assigned the Membership-Domain of "Product X" and Membership-Value of 1. User 2 is assigned a Membership-Domain of "Product X" and a Membership-Value of 2. User 3 is assigned a Membership-Domain of "Product X" and Membership-Value of 4.

7.2. Rule in multiple sets (1 Domain)

Expanding upon our example from above Section 7.1, a new Filter-Rule is added that shall be part of Users with either Rule Set 1 or Rule Set 2 of Product X.

Accordingly, the policy administrator defines the Membership-Assignment having a Membership-Domain of "Product X", a Membership-Value of 3 and a Match-Type of OVERLAP. Thus, any Policy-Membership whose Membership-Value is set to 1 or 2 will have this Filter-Rule applied.

7.3. Default Route (Overlapping) Rules

A common traffic rule is the default (all traffic) rule. It is often used as the lowest priority rule in a policy enforcement session. Even though the rule is typically the same, e.g. "any any", the actions taken may vary, e.g. deny traffic, permit traffic, set quality of service. To distinguish the rules the use of the

Membership-Domain in the Membership-Assignment even when the Membership-Value MAY be the same.

Within the enforcement point, for each overlapping Match-Type can be set to OVERLAP and contain all bits where the rule applies in its Membership-Value. In general, the Membership-Value MUST be NOT overlap with other default rules or a Precedence MUST be followed.

In the case where a Filter-Rule [RFC5777] is used, the Match-Type and Membership-Value can be used as part of the Classifier AVP.

8. IANA Considerations

IANA allocated AVP codes in the IANA-controlled namespace registry specified in Section 11.1.1 of [RFC6733] for the following AVPs that are defined in this document.

AVP	AVP Code	Section Defined	Data Type
Base-Name	TBD1	Section 5.1	UTF8String
Policy-Membership	TBD2	Section 5.2	GROUPED
Membership-Assignment	TBD3	Section 5.3	GROUPED
Membership-Domain	TBD4	Section 5.4	UTF8String
Membership-Value	TBD5	Section 5.5	OctetString
Match-Type	TBD6	Section 5.6	Enumerated

9. Security Considerations

The use of Base-Names and Membership-Domain can unintentionally provide user information if it is too explicit, e.g. "Bobs' Policies". It is RECOMMENDED that an operator consider the values it assigns and ensure they provide no user or group specific information.

As bit and test patterns the data provided by the Membership-Assignment and Policy-Membership AVPs provide more clues between an Operator and Authorized User's policy relationship. However, it is no different than if one has access to the information transmitted between the Diameter Client and Server today (if the Base-Names and Membership-Domains) follow the recommendations in this section.

In either case, access to the Diameter communications is still required.

The Security Considerations of the Diameter protocol itself have been discussed in [RFC6733]. The Diameter base protocol [RFC6733] requires that each Diameter implementation use underlying security; i.e., TLS/TCP, DTLS/SCTP or IPsec. Use of the AVPs defined in this document MUST take into consideration the security issues and requirements of the Diameter base protocol.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.

10.2. Informative References

- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, DOI 10.17487/RFC5777, February 2010, <<http://www.rfc-editor.org/info/rfc5777>>.
- [RFC7155] Zorn, G., Ed., "Diameter Network Access Server Application", RFC 7155, DOI 10.17487/RFC7155, April 2014, <<http://www.rfc-editor.org/info/rfc7155>>.

Authors' Addresses

Lyle Bertz
Sprint
6220 Sprint Parkway
Overland Park, KS 66251
United States

Email: lylebe551144@gmail.com

Mark Bales
Sprint
6220 Sprint Parkway
Overland Park, KS 66251
United States

Email: yellowjeep2017@gmail.com

Diameter Maintenance and Extensions
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2017

L. Bertz
Sprint
June 29, 2017

Diameter Predicted Units
draft-bertz-dime-predictunits-02

Abstract

This document specifies the conveyance of predicted usage information for proper dimensioning of network services that use Diameter based authorization.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Predicted Service AVPs	3
3.1. Predicted-Service-Units	3
3.2. Predicted-Service-Units-Series	4
4. Usage Examples	5
5. IANA Considerations	5
6. Security Considerations	5
7. References	6
7.1. Normative References	6
7.2. Informative References	6
Author's Address	7

1. Introduction

When a User is authorized to use a service via Diameter applications such as [RFC4006] or [RFC7155], the Client is not aware of the average load placed upon it by the User. This can lead to overload situations or Diameter Clients being too conservative and denying services to valid Users even whose presence would not overload the service.

Given virtualization and the use of many software based services the service capacity varies on a service instance, i.e. Diameter Client, basis. Even though the Diameter Client is the same software it will vary in terms of the load it can accept. Thus, a Diameter Server cannot depend upon consistent capacities of a Diameter Client.

This specification introduces the Predicted-Service-Units Attribute Value Pair (AVP). This information conveys the predicted usage introduced on the service by the authorized User. Such information can be used by the Diameter Client to estimate future load and proactively manage its resources.

Although this information is conveyed from the Diameter Server to the Client several system aspects are out of the scope of this document:

- o How the Diameter Server acquired the information contained in the Predicted-Service-Units AVP.
- o How the values in the Predicted-Service-Units AVP were determined.
- o The accuracy or validity of the values in the Predicted-Service-Units AVP.
- o Specific actions the Diameter Client should take when its service functions are overloaded or are predicted to be overloaded based upon the information provided by Predicted-Service-Units.
- o Specific actions the Diameter Client takes to bring itself in/out of service for new or existing Users.

When the value(s) or multiple types of Costs are provided they are represented by the Time-Of-Day-Condition AVP defined in [RFC5777] and contained in a Predicted-Service-Units-Series AVP. This AVP contains one or more Predicted-Service-Units. Multiple Cost types, e.g. CC-Total-Octets and CC-Time, may be represented in the same Predicted-Service-Units entry and in the same Predicted-Service-Units-Series so long as no overlapping times exist for the same Cost Type.

2. Terminology

In this document, the key words "MAY", "MUST", "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119].

3. Predicted Service AVPs

3.1. Predicted-Service-Units

The Predicted-Service-Units AVP (AVP Code TBD1) is of type Grouped and contains the amount of units that the Diameter Client can expect to provide to the end user until the service must be released or the new service authorization request, e.g. Credit-Control-Request, must be sent if a Granted-Service-Unit AVP [RFC4006] has been applied to the user's service. A client is not required to implement all of the unit types, and it MUST ignore unknown or unsupported unit types.

The Predicted-Service-Units AVP is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Predicted-Service-Units ::= < AVP Header: TBD1 >
    [ CC-Time ]
    [ CC-Money ]
    [ CC-Total-Octets ]
    [ CC-Input-Octets ]
    [ CC-Output-Octets ]
    [ CC-Service-Specific-Units ]
    [ Time-Of-Day-Condition ]
    *[ AVP ]
```

The Time-Of-Day-Condition AVP is defined in [RFC5777], all other AVPs are defined in [RFC4006].

The presence of this information is provided as anticipated load information to the Diameter Client and is not intended to be prescriptive in any manner regarding the user's service.

When the Time-Of-Day-Condition AVP is not present, the value(s) are assumed to apply for the duration of the authorized session until this value is updated as part of the Diameter application, e.g. a Diameter Re-Auth-Request/Answer (RAR/RAA) message [RFC6733].

3.2. Predicted-Service-Units-Series

The Predicted-Service-Units-Series AVP (AVP Code TBD2) is of type Grouped, and contains one or more Predicted-Service-Units with non-overlapping times for each specific Cost type.

A client is not required to implement all of the unit types, and it MUST ignore unknown or unsupported unit types.

It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Predicted-Service-Units-Series ::= < AVP Header: TBD2 >
    1*{ Predicted-Service-Units }
```

For each specific type of Cost, e.g. CC-Time, any two Predicted-Service-Units values in the series MUST NOT contain overlapping time windows specified in their Time-Of-Day-Condition values. When an entry has no Time-Of-Day-Condition present it is assumed to apply at all times.

4. Usage Examples

When Predicted-Service-Units are returned as part of an authorization per [RFC7155] or [RFC4006], the client MAY use this information as guidance on projected load the new user will generate on the service.

If the client supports/understands the information provided in the Predicted-Service-Units AVP, it can update its projected load. Based upon this information it MAY take one or more of the following actions (this is not exhaustive):

- o Redirect any new service requests at the service / protocol level.
- o Begin enforcing mechanisms to reduce the amount of service load on a subset of services already established.
- o Remove itself from any system that directs new service requests to it.
- o Initiate administrative functions to increase its capacity or start the process of creating new instances to service future requests.

5. IANA Considerations

IANA allocated AVP codes in the IANA-controlled namespace registry specified in Section 11.1.1 of [RFC6733] for the following AVPs that are defined in this document.

AVP	AVP Code	Section Defined	Data Type
Predicted-Service-Units	TBD1	Section 3.1	GROUPED
Predicted-Service-Units-Series	TBD2	Section 3.2	GROUPED

6. Security Considerations

The Diameter base protocol [RFC6733] requires that each Diameter implementation use underlying security; i.e., TLS/TCP, DTLS/SCTP or IPsec. 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 [RFC6733]. In these environments, the use of TLS/TCP, DTLS/SCTP or IPsec is sufficient. The details of TLS/TCP, DTLS/SCTP or IPsec related security considerations are discussed in the [RFC6733].

Because this application conveys past usage information (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; e.g., 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.

The attributes provided by this solution MUST be assumed to be privacy sensitive by both the client and server.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, "Diameter Credit-Control Application", RFC 4006, DOI 10.17487/RFC4006, August 2005, <<http://www.rfc-editor.org/info/rfc4006>>.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, DOI 10.17487/RFC5777, February 2010, <<http://www.rfc-editor.org/info/rfc5777>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.

7.2. Informative References

- [RFC7155] Zorn, G., Ed., "Diameter Network Access Server Application", RFC 7155, DOI 10.17487/RFC7155, April 2014, <<http://www.rfc-editor.org/info/rfc7155>>.

Author's Address

Lyle Bertz
Sprint
6220 Sprint Parkway
Overland Park, KS 66251
United States

Email: lylebe551144@gmail.com

Diameter Maintenance and Extensions (DIME)
Internet-Draft
Updates: RFC7683 (if approved)
Intended status: Standards Track
Expires: September 8, 2017

S. Donovan
Oracle
March 7, 2017

Diameter Agent Overload and the Peer Overload Report
draft-ietf-dime-agent-overload-10.txt

Abstract

This specification documents an extension to RFC 7683 (Diameter Overload Indication Conveyance (DOIC)) base solution. The extension defines the Peer overload report type. The initial use case for the Peer report is the handling of occurrences of overload of a Diameter agent.

Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 8, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- 1. Introduction 3
- 2. Terminology and Abbreviations 3
- 3. Peer Report Use Cases 4
 - 3.1. Diameter Agent Overload Use Cases 4
 - 3.1.1. Single Agent 4
 - 3.1.2. Redundant Agents 5
 - 3.1.3. Agent Chains 7
 - 3.2. Diameter Endpoint Use Cases 7
 - 3.2.1. Hop-by-hop Abatement Algorithms 8
- 4. Interaction Between Host/Realm and Peer Overload Reports . . 8
- 5. Peer Report Behavior 8
 - 5.1. Capability Announcement 8
 - 5.1.1. Reacting Node Behavior 8
 - 5.1.2. Reporting Node Behavior 9
 - 5.2. Peer Overload Report Handling 10
 - 5.2.1. Overload Control State 10
 - 5.2.2. Reporting Node Maintenance of Peer Report OCS 11
 - 5.2.3. Reacting Node Maintenance of Peer Report OCS 11
 - 5.2.4. Peer-Report Reporting Node Behavior 12
 - 5.2.5. Peer-Report Reacting Node Behavior 12
- 6. Peer Report AVPs 13
 - 6.1. OC-Supported-Features AVP 13
 - 6.1.1. OC-Feature-Vector AVP 14
 - 6.1.2. OC-Peer-Algo AVP 14
 - 6.2. OC-OLR AVP 14
 - 6.2.1. OC-Report-Type AVP 15
 - 6.3. SourceID AVP 15
 - 6.4. Attribute Value Pair Flag Rules 15
- 7. IANA Considerations 16
 - 7.1. AVP Codes 16
 - 7.2. New Registries 16
- 8. Security Considerations 16
- 9. Acknowledgements 17
- 10. References 17
 - 10.1. Informative References 17
 - 10.2. Normative References 17
- Author's Address 17

1. Introduction

This specification documents an extension to the Diameter Overload Indication Conveyance (DOIC) [RFC7683] base solution. The extension defines the Peer overload report type. The initial use case for the Peer report is the handling of occurrences of overload of a Diameter agent.

This document defines the behavior of Diameter nodes when Diameter agents enter an overload condition and send an overload report requesting a reduction of traffic. It also defines new overload report type, the Peer overload report type, that is used for handling of agent overload conditions. The Peer overload report type is defined in a generic fashion so that it can also be used for other Diameter overload scenarios.

The base Diameter overload specification [RFC7683] addresses the handling of overload when a Diameter endpoint (a Diameter Client or Diameter Server as defined in [RFC6733]) becomes overloaded.

In the base specification, the goal is to handle abatement of the overload occurrence as close to the source of the Diameter traffic as feasible. When possible this is done at the originator of the traffic, generally referred to as a Diameter Client. A Diameter Agent might also handle the overload mitigation. For instance, a Diameter Agent might handle Diameter overload mitigation when it knows that a Diameter Client does not support the DOIC extension.

This document extends the base Diameter endpoint overload specification to address the case when Diameter Agents become overloaded. Just as is the case with other Diameter nodes -- Diameter Clients and Diameter Servers -- surges in Diameter traffic can cause a Diameter Agent to be asked to handle more Diameter traffic than it was configured to handle. For a more detailed discussion of what can cause the overload of Diameter nodes, refer to the Diameter Overload Requirements [RFC7068].

This document defines a new overload report type to communicate occurrences of agent overload. This report type works for the "Loss" overload mitigation algorithm defined in [RFC7683] and is expected to work for other overload abatement algorithms defined in extensions to the DOIC solution.

2. Terminology and Abbreviations

Diameter Node

A RFC6733 Diameter Client, an RFC6733 Diameter Server, and RFC6733 Diameter Agent.

Diameter Endpoint

An RFC6733 Diameter Client and RFC6733 Diameter Server.

Diameter Agent

An RFC6733 Diameter Agent.

Reporting Node

A DOIC Node that sends an overload report in a Diameter answer message.

Reacting Node

A DOIC Node that receives and acts on a DOIC overload report.

DOIC Node

A Diameter Node that supports the DOIC solution defined in [RFC7683].

3. Peer Report Use Cases

This section outlines representative use cases for the peer report used to communicate agent overload.

There are two primary classes of use cases currently identified, those involving the overload of agents and those involving overload of Diameter endpoints. In both cases the goal is to use an overload algorithm that controls traffic sent towards peers.

3.1. Diameter Agent Overload Use Cases

The peer report needs to support the following use cases.

3.1.1. Single Agent

This use case is illustrated in Figure 1. In this case, the client sends all traffic through the single agent. If there is a failure in the agent then the client is unable to send Diameter traffic toward the server.

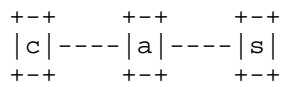


Figure 1

A more likely case for the use of agents is illustrated in Figure 2. In this case, there are multiple servers behind the single agent. The client sends all traffic through the agent and the agent determines how to distribute the traffic to the servers based on local routing and load distribution policy.

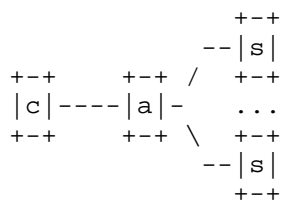


Figure 2

In both of these cases, the occurrence of overload in the single agent must be handled by the client in a similar fashion as if the client were handling the overload of a directly connected server. When the agent becomes overloaded it will insert an overload report in answer messages flowing to the client. This overload report will contain a requested reduction in the amount of traffic sent to the agent. The client will apply overload abatement behavior as defined in the base Diameter overload specification [RFC7683] or the extension draft that defines the indicated overload abatement algorithm. This will result in the throttling of the abated traffic that would have been sent to the agent, as there is no alternative route. The client sends an appropriate error response to the originator of the request.

3.1.2. Redundant Agents

Figure 3 and Figure 4 illustrate a second, and more likely, type of deployment scenario involving agents. In both of these cases, the client has Diameter connections to two agents.

Figure 3 illustrates a client that has a primary connection to one of the agents (agent a1) and a secondary connection to the other agent (agent a2). In this scenario, under normal circumstances, the client

will use the primary connection for all traffic. The secondary connection is used when there is a failure scenario of some sort.

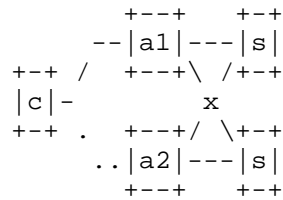


Figure 3

The second case, in Figure 4, illustrates the case where the connections to the agents are both actively used. In this case, the client will have local distribution policy to determine the traffic sent through each client.

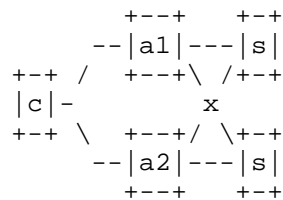


Figure 4

In the case where one of the agents in the above scenario becomes overloaded, the client should reduce the amount of traffic sent to the overloaded agent by the amount requested. This traffic should instead be routed through the non-overloaded agent. For example, assume that the overloaded agent requests a reduction of 10 percent. The client should send 10 percent of the traffic that would have been routed to the overloaded agent through the non-overloaded agent.

When the client has an active and a standby connection to the two agents then an alternative strategy for responding to an overload report from an agent is to change the standby connection to active and route all traffic through the new active connection.

In the case where both agents are reporting overload, the client may need to start decreasing the total traffic sent to the agents. This would be done in a similar fashion as discussed in Section 3.1.1 The amount of traffic depends on the combined reduction requested by the two agents.

3.1.3. Agent Chains

There are also deployment scenarios where there can be multiple Diameter Agents between Diameter Clients and Diameter Servers. An example of this type of deployment includes when there are Diameter agents between administrative domains.

Figure 5 illustrates one such network deployment case. Note that while this figure shows a maximum of two agents being involved in a Diameter transaction, it is possible that more than two agents could be in the path of a transaction.

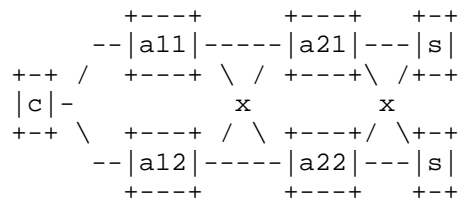


Figure 5

Handling of overload of one or both of agents a11 or a12 in this case is equivalent to that discussed in Section 3.1.2.

Overload of agents a21 and a22 must be handled by the previous hop agents. As such, agents a11 and a12 must handle the overload mitigation logic when receiving an agent overload report from agents a21 and a22.

The handling of peer overload reports is similar to that discussed in Section 3.1.2. If the overload can be addressed using diversion then this approach should be taken.

If both of the agents have requested a reduction in traffic then the previous hop agent must start throttling the appropriate number of transactions. When throttling requests, an agent uses the same error responses as defined in the base DOIC specification [RFC7683].

3.2. Diameter Endpoint Use Cases

This section outlines use cases for the peer overload report involving Diameter Clients and Diameter Servers.

3.2.1. Hop-by-hop Abatement Algorithms

It is envisioned that abatement algorithms will be defined that will support the option for Diameter Endpoints to send peer reports. For instance, it is envisioned that one usage scenario for the rate algorithm, [I-D.ietf-dime-doic-rate-control], which is being worked on by the DIME working group as this document is being written, will involve abatement being done on a hop-by-hop basis.

This rate deployment scenario would involve Diameter Endpoints generating peer reports and selecting the rate algorithm for abatement of overload conditions.

4. Interaction Between Host/Realm and Peer Overload Reports

It is possible that both an agent and an end-point in the path of a transaction are overloaded at the same time. When this occurs, Diameter entities need to handle both overload reports. In this scenario the reacting node should first handle the throttling of the overloaded host or realm. Any messages that survive throttling due to host or realm reports should then go through abatement for the peer overload report. In this scenario, when doing abatement on the PEER report, the reacting node SHOULD take into consideration the number of messages already throttled by the handling of the HOST/REALM report abatement.

Note: The goal is to avoid traffic oscillations that might result from throttling of messages for both the HOST/REALM overload reports and the PEER overload reports. This is especially a concern if both reports indicate the LOSS abatement algorithm.

5. Peer Report Behavior

This section defines the normative behavior associated with the Peer Report extension to the DOIC solution.

5.1. Capability Announcement

5.1.1. Reacting Node Behavior

When sending a Diameter request a DOIC Node that supports the OC_PEER_REPORT (as defined in Section 6.1.1) feature MUST include in the OC-Supported-Features AVP an OC-Feature-Vector AVP with the OC_PEER_REPORT bit set.

When sending a request a DOIC Node that supports the OC_PEER_REPORT feature MUST include a SourceID AVP in the OC-Supported-Features AVP with its own DiameterIdentity.

When a Diameter Agent relays a request that includes a SourceID AVP in the OC-Supported-Features AVP, if the Diameter Agent supports the OC_PEER_REPORT feature then it MUST remove the received SourceID AVP and replace it with a SourceID AVP containing its own DiameterIdentity.

5.1.2. Reporting Node Behavior

When receiving a request a DOIC Node that supports the OC_PEER_REPORT feature MUST update transaction state with an indication of whether or not the peer from which the request was received supports the OC_PEER_REPORT feature.

Note: The transaction state is used when the DOIC Node is acting as a peer-report reporting node and needs send OC-OLR reports of type peer in answer messages. The peer overload reports are only included in answer messages being sent to peers that support the OC_PEER_REPORT feature.

The peer supports the OC_PEER_REPORT feature if the received request contains an OC-Supported-Features AVP with the OC-Feature-Vector with the OC_PEER_REPORT feature bit set and with a SourceID AVP with a value that matches the DiameterIdentity of the peer from which the request was received.

When an agent relays an answer message, a reporting node that supports the OC_PEER_REPORT feature MUST strip any SourceID AVP from the OC-Supported-Features AVP.

When sending an answer message, a reporting node that supports the OC_PEER_REPORT feature MUST determine if the peer to which the answer is to be sent supports the OC_PEER_REPORT feature.

If the peer supports the OC_PEER_REPORT feature then the reporting node MUST indicate support for the feature in the OC-Supported-Features AVP.

If the peer supports the OC_PEER_REPORT feature then the reporting node MUST insert the SourceID AVP in the OC-Supported-Features AVP in the answer message.

If the peer supports the OC_PEER_REPORT feature then the reporting node MUST insert the OC-Peer-Algo AVP in the OC-Supported-Features AVP. The OC-Peer-Algo AVP MUST indicate the overload abatement algorithm that the reporting node wants the reacting nodes to use should the reporting node send a peer overload report as a result of becoming overloaded.

5.2. Peer Overload Report Handling

This section defines the behavior for the handling of overload reports of type peer.

5.2.1. Overload Control State

This section describes the Overload Control State (OCS) that might be maintained by both the peer-report reporting node and the peer-report reacting node.

This is an extension of the OCS handling defined in [RFC7683].

5.2.1.1. Reporting Node Peer Report OCS

A DOIC Node that supports the OC_PEER_REPORT feature SHOULD maintain Reporting Node OCS, as defined in [RFC7683] and extended here.

If different abatement specific contents are sent to each peer then the reporting node MUST maintain a separate reporting node peer report OCS entry per peer to which a peer overload report is sent.

Note: The rate overload abatement algorithm allows for different rates to be sent to each peer.

5.2.1.2. Reacting Node Peer Report OCS

In addition to OCS maintained as defined in [RFC7683], a reacting node that supports the OC_PEER_REPORT feature maintains the following OCS per supported Diameter application:

A peer-type OCS entry for each peer to which it sends requests.

A peer-type OCS entry is identified by the pair of Application-ID and the peer's DiameterIdentity.

The peer-type OCS entry include the following information (the actual information stored is an implementation decision):

Sequence number (as received in the OC-OLR AVP).

Time of expiry (derived from OC-Validity-Duration AVP received in the OC-OLR AVP and time of reception of the message carrying OC-OLR AVP).

Selected abatement algorithm (as received in the OC-Supported-Features AVP).

Input data that is abatement algorithm specific (as received in the OC-OLR AVP -- for example, OC-Reduction-Percentage for the loss abatement algorithm).

5.2.2. Reporting Node Maintenance of Peer Report OCS

All rules for managing the reporting node OCS entries defined in [RFC7683] apply to the peer report.

5.2.3. Reacting Node Maintenance of Peer Report OCS

When a reacting node receives an OC-OLR AVP with a report type of peer it MUST determine if the report was generated by the Diameter peer from which the report was received.

If a reacting node receives an OC-OLR AVP of type peer and the SourceID matches the DiameterIdentity of the Diameter peer from which the response message was received then the report was generated by a Diameter peer.

If a reacting node receives an OC-OLR AVP of type peer and the SourceID does not match the DiameterIdentity of the Diameter peer from which the response message was received then the reacting node MUST ignore the overload report.

Note: Under normal circumstances, a Diameter node will not add a peer report when sending to a peer that does not support this extension. This requirement is to handle the case where peer reports are erroneously or maliciously inserted into response messages.

If the peer report was received from a Diameter peer then the reacting node MUST determine if it is for an existing or new overload condition.

The peer report is for an existing overload condition if the reacting node has an OCS that matches the received peer report. For a peer report, this means it matches the Application-ID and the peer's DiameterIdentity in an existing OCS entry.

If the peer report is for an existing overload condition then it MUST determine if the peer report is a retransmission or an update to the existing OLR.

If the sequence number for the received peer report is greater than the sequence number stored in the matching OCS entry then the reacting node MUST update the matching OCS entry.

If the sequence number for the received peer report is less than or equal to the sequence number in the matching OCS entry then the reacting node MUST silently ignore the received peer report. The matching OCS MUST NOT be updated in this case.

If the received peer report is for a new overload condition then the reacting node MUST generate a new OCS entry for the overload condition.

For a peer report this means it creates an OCS entry with a DiameterIdentity from the SourceID AVP in the received OC-OLR AVP.

If the received peer report contains a validity duration of zero ("0") then the reacting node MUST update the OCS entry as being expired.

The reacting node does not delete an OCS when receiving an answer message that does not contain an OC-OLR AVP (i.e. absence of OLR means "no change").

The reacting node sets the abatement algorithm based on the OC-Peer-Algo AVP in the received OC-Supported-Features AVP.

5.2.4. Peer-Report Reporting Node Behavior

When there is an existing reporting node peer report OCS entry, the reporting node MUST include an OC-OLR AVP with a report type of peer using the contents of the reporting node peer report OCS entry in all answer messages sent by the reporting node to peers that support the OC_PEER_REPORT feature.

The reporting node determines if a peer supports the OC_PEER_REPORT feature based on the indication recorded in the reporting node's transaction state.

The reporting node MUST include its DiameterIdentity in the SourceID AVP in the OC-OLR AVP. This is used by DOIC Nodes that support the OC_PEER_REPORT feature to determine if the report was received from a Diameter peer.

The reporting agent must follow all other overload reporting node behaviors outlined in the DOIC specification.

5.2.5. Peer-Report Reacting Node Behavior

A reacting node supporting this extension MUST support the receipt of multiple overload reports in a single message. The message might

include a host overload report, a realm overload report and/or a peer overload report.

When a reacting node sends a request it MUST determine if that request matches an active OCS.

In all cases, if the reacting node is an agent then it MUST strip the Peer Report OC-OLR AVP from the message.

If the request matches an active OCS then the reacting node MUST apply abatement treatment to the request. The abatement treatment applied depends on the abatement algorithm indicated in the OCS.

For peer overload reports, the preferred abatement treatment is diversion. As such, the reacting node SHOULD attempt to divert requests identified as needing abatement to other peers.

If there is not sufficient capacity to divert abated traffic then the reacting node MUST throttle the necessary requests to fit within the available capacity of the peers able to handle the requests.

If the abatement treatment results in throttling of the request and if the reacting node is an agent then the agent MUST send an appropriate error response as defined in [RFC7683].

In the case that the OCS entry validity duration expires or has a validity duration of zero ("0"), meaning that if the reporting node has explicitly signaled the end of the overload condition then abatement associated with the OCS entry MUST be ended in a controlled fashion.

6. Peer Report AVPs

6.1. OC-Supported-Features AVP

This extension adds a new feature to the OC-Feature-Vector AVP. This feature indication shows support for handling of peer overload reports. Peer overload reports are used by agents to indicate the need for overload abatement handling by the agent's peer.

A supporting node must also include the SourceID AVP in the OC-Supported-Features capability AVP.

This AVP contains the DiameterIdentity of the node that supports the OC_PEER_REPORT feature. This AVP is used to determine if support for the peer overload report is in an adjacent node. The value of this AVP should be the same Diameter identity used as part of the Diameter Capabilities Exchange procedure defined in [RFC7683].

This extension also adds the OC-Peer-Algo AVP to the OC-Supported-Features AVP. This AVP is used by a reporting node to indicate the abatement algorithm it will use for peer overload reports.

```
OC-Supported-Features ::= < AVP Header: 621 >
                        [ OC-Feature-Vector ]
                        [ SourceID ]
                        [ OC-Peer-Algo]
                        * [ AVP ]
```

6.1.1. OC-Feature-Vector AVP

The peer report feature defines a new feature bit for the OC-Feature-Vector AVP.

OC_PEER_REPORT (0x0000000000000010)

When this flag is set by a DOIC Node it indicates that the DOIC Node supports the peer overload report type.

6.1.2. OC-Peer-Algo AVP

The OC-Peer-Algo AVP (AVP code TBD1) is of type Unsigned64 and contains a 64 bit flags field of announced capabilities of a DOIC Node. The value of zero (0) is reserved.

Feature bits defined for the OC-Feature-Vector AVP and associated with overload abatement algorithms are reused for this AVP.

6.2. OC-OLR AVP

This extension makes no changes to the OC_Sequence_Number or OC_Validity_Duration AVPs in the OC-OLR AVP. These AVPs are also be used in peer overload reports.

The OC_PEER_REPORT feature extends the base Diameter overload specification by defining a new overload report type of "peer". See section [7.6] in [RFC7683] for a description of the OC-Report-Type AVP.

The overload report MUST also include the Diameter identity of the agent that generated the report. This is necessary to handle the case where there is a non supporting agent between the reporting node and the reacting node. Without the indication of the agent that generated the overload report, the reacting node could erroneously assume that the report applied to the non-supporting node. This

could, in turn, result in unnecessary traffic being either diverted or throttled.

The SourceID AVP is used in the OC-OLR AVP to carry this DiameterIdentity.

```
OC-OLR ::= < AVP Header: 623 >
         < OC-Sequence-Number >
         < OC-Report-Type >
         [ OC-Reduction-Percentage ]
         [ OC-Validity-Duration ]
         [ SourceID ]
         * [ AVP ]
```

6.2.1. OC-Report-Type AVP

The following new report type is defined for the OC-Report-Type AVP.

PEER_REPORT 2 The overload treatment should apply to all requests bound for the peer identified in the overload report. If the peer identified in the overload report is not a peer to the reacting endpoint then the overload report should be stripped and not acted upon.

6.3. SourceID AVP

The SourceID AVP (AVP code TBD2) is of type DiameterIdentity and is inserted by a Diameter node to indicate the source of the AVP in which it is a part.

In the case of peer reports, the SourceID AVP indicates the node that supports this feature (in the OC-Supported-Features AVP) or the node that generates an overload with a report type of peer (in the OC-OLR AVP).

It contains the DiameterIdentity of the inserting node. This is used by other Diameter nodes to determine the node that inserted the enclosing AVP that contains the SourceID AVP.

6.4. Attribute Value Pair Flag Rules

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules	
				MUST	NOT
OC-Peer-Algo	TBD1	6.1.2	Unsigned64	V	
SourceID	TBD2	6.3	DiameterIdentity	V	

7. IANA Considerations

7.1. AVP Codes

New AVPs defined by this specification are listed in Section 6.4. All AVP codes are allocated from the 'Authentication, Authorization, and Accounting (AAA) Parameters' AVP Codes registry.

One new OC-Report-Type AVP value is defined in Section 6.2.1

7.2. New Registries

There are no new IANA registries introduced by this document.

The values used for the OC-Peer-Algo AVP are the subset of the "OC-Feature-Vector AVP Values (code 622)" registry. Only the values in that registry that apply to overload abatement algorithms apply to the OC-Peer-Algo AVP.

8. Security Considerations

Agent overload is an extension to the base Diameter overload mechanism. As such, all of the security considerations outlined in [RFC7683] apply to the agent overload scenarios.

It is possible that the malicious insertion of an agent overload report could have a bigger impact on a Diameter network as agents can be concentration points in a Diameter network. Where an end-point report would impact the traffic sent to a single Diameter server, for example, a peer report could throttle all traffic to the Diameter network.

This impact is amplified in an agent that sits at the edge of a Diameter network that serves as the entry point from all other Diameter networks.

9. Acknowledgements

Adam Roach and Eric McMurry for the work done in defining a comprehensive Diameter overload solution in draft-roach-dime-overload-ctrl-03.txt.

Ben Campbell for his insights and review of early versions of this document.

10. References

10.1. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7068] McMurry, E. and B. Campbell, "Diameter Overload Control Requirements", RFC 7068, DOI 10.17487/RFC7068, November 2013, <<http://www.rfc-editor.org/info/rfc7068>>.

10.2. Normative References

- [I-D.ietf-dime-doic-rate-control] Donovan, S. and E. Noel, "Diameter Overload Rate Control", draft-ietf-dime-doic-rate-control-03 (work in progress), March 2016.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.
- [RFC7683] Korhonen, J., Ed., Donovan, S., Ed., Campbell, B., and L. Morand, "Diameter Overload Indication Conveyance", RFC 7683, DOI 10.17487/RFC7683, October 2015, <<http://www.rfc-editor.org/info/rfc7683>>.

Author's Address

Steve Donovan
Oracle
7460 Warren Parkway, Suite 300
Frisco, Texas 75034
United States

Email: srdonovan@usdonovans.com

Diameter Maintenance and Extensions (DIME)
Internet-Draft
Intended status: Standards Track
Expires: August 20, 2017

S. Donovan, Ed.
Oracle
E. Noel
AT&T Labs
February 16, 2017

Diameter Overload Rate Control
draft-ietf-dime-doic-rate-control-05.txt

Abstract

This specification documents an extension to the Diameter Overload Indication Conveyance (DOIC) [RFC7683] base solution. This extension adds a new overload control abatement algorithm. This abatement algorithm allows for a DOIC reporting node to specify a maximum rate at which a DOIC reacting node sends Diameter requests to the DOIC reporting node.

Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 20, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology and Abbreviations	4
3. Interaction with DOIC report types	5
4. Capability Announcement	5
5. Overload Report Handling	6
5.1. Reporting Node Overload Control State	6
5.2. Reacting Node Overload Control State	6
5.3. Reporting Node Maintenance of Overload Control State	7
5.4. Reacting Node Maintenance of Overload Control State	7
5.5. Reporting Node Behavior for Rate Abatement Algorithm	7
5.6. Reacting Node Behavior for Rate Abatement Algorithm	8
6. Rate Abatement Algorithm AVPs	8
6.1. OC-Supported-Features AVP	8
6.1.1. OC-Feature-Vector AVP	8
6.2. OC-OLR AVP	9
6.2.1. OC-Maximum-Rate AVP	9
6.3. Attribute Value Pair flag rules	9
7. Rate Based Abatement Algorithm	10
7.1. Overview	10
7.2. Reporting Node Behavior	10
7.3. Reacting Node Behavior	11
7.3.1. Default Algorithm	11
7.3.2. Priority Treatment	14
7.3.3. Optional Enhancement: Avoidance of Resonance	16
8. IANA Consideration	17
8.1. AVP codes	17
8.2. New registries	17
9. Security Considerations	17
10. Acknowledgements	18
11. References	18
11.1. Normative References	18
11.2. Informative References	18
Authors' Addresses	18

1. Introduction

This document defines a new Diameter overload control abatement algorithm.

The base Diameter overload specification [RFC7683] defines the loss algorithm as the default Diameter overload abatement algorithm. The loss algorithm allows a reporting node to instruct a reacting node to reduce the amount of traffic sent to the reporting node by abating (diverting or throttling) a percentage of requests sent to the server. While this can effectively decrease the load handled by the server, it does not directly address cases where the rate of arrival of service requests increases quickly. If the service requests that result in Diameter transactions increase quickly then the loss algorithm cannot guarantee the load presented to the server remains below a specific rate level. The loss algorithm can be slow to protect the stability of reporting nodes when subjected with rapidly changing loads.

Consider the case where a reacting node is handling 100 service requests per second, where each of these service requests results in one Diameter transaction being sent to a reporting node. If the reporting node is approaching an overload state, or is already in an overload state, it will send a Diameter overload report requesting a percentage reduction in traffic sent. Assume for this discussion that the reporting node requests a 10% reduction. The reacting node will then abate (diverting or throttling) ten Diameter transactions a second, sending the remaining 90 transactions per second to the reporting node.

Now assume that the reacting node's service requests spikes to 1000 requests per second. The reacting node will continue to honor the reporting node's request for a 10% reduction in traffic. This results, in this example, in the reacting node sending 900 Diameter transactions per second, abating the remaining 100 transactions per second. This spike in traffic is significantly higher than the reporting node is expecting to handle and can result in negative impacts to the stability of the reporting node.

The reporting node can, and likely would, send another overload report requesting that the reacting node abate 91% of requests to get back to the desired 90 transactions per second. However, once the spike has abated and the reacting node handled service requests returns to 100 per second, this will result in just 9 transactions per second being sent to the reporting node, requiring a new overload report setting the reduction percentage back to 10%. This control feedback loop has the potential to make the situation worse by

causing wide fluctuations in traffic on multiple nodes in the Diameter network.

One of the benefits of a rate based algorithm is that it better handles spikes in traffic. Instead of sending a request to reduce traffic by a percentage, the rate approach allows the reporting node to specify the maximum number of Diameter requests per second that can be sent to the reporting node. For instance, in this example, the reporting node could send a rate-based request specifying the maximum transactions per second to be 90. The reacting node will send the 90 regardless of whether it is receiving 100 or 1000 service requests per second.

This document extends the base DOIC solution [RFC7683] to add support for the rate based overload abatement algorithm.

This document draws heavily on work in the SIP Overload Control working group. The definition of the rate abatement algorithm is copied almost verbatim from the SOC document [RFC7415], with changes focused on making the wording consistent with the DOIC solution and the Diameter protocol.

2. Terminology and Abbreviations

Diameter Node

A RFC6733 Diameter Client, RFC6733 Diameter Server, or RFC6733 Diameter Agent.

Diameter Endpoint

An RFC6733 Diameter Client or RFC6733 Diameter Server.

DOIC Node

A Diameter Node that supports the DOIC solution defined in [RFC7683].

Reporting Node

A DOIC Node that sends a DOIC overload report.

Reacting Node

A DOIC Node that receives and acts on a DOIC overload report.

3. Interaction with DOIC report types

As of the publication of this specification there are two DOIC report types defined with the specification of a third in progress:

1. Host - Overload of a specific Diameter Application at a specific Diameter Node as defined in [RFC7683].
2. Realm - Overload of a specific Diameter Application at a specific Diameter Realm as defined in [RFC7683].
3. Peer - Overload of a specific Diameter peer as defined in [I-D.ietf-dime-agent-overload].

The rate algorithm MAY be selected by reporting nodes for any of these report types.

It is expected that all report types defined in the future will indicate whether or not the rate algorithm can be used with that report type.

4. Capability Announcement

This extension defines the rate abatement algorithm (referred to as rate in this document) feature. Support for the rate feature will be reflected by use of a new value, as defined in Section 6.1.1, in the OC-Feature-Vector AVP per the rules defined in [RFC7683].

Note that Diameter nodes that support the rate feature will, by definition, support both the loss and rate based abatement algorithms. DOIC reacting nodes SHOULD indicate support for both the loss and rate algorithms in the OC-Feature-Vector AVP.

There may be local policy reasons that cause a DOIC node that supports the rate abatement algorithm to not include it in the OC-Feature-Vector. All reacting nodes, however, must continue to include loss in the OC-Feature-Vector in order to remain compliant with [RFC7683].

A reporting node MAY select one abatement algorithm to apply to host and realm reports and a different algorithm to apply to peer reports.

For host or realm reports the selected algorithm is reflected in the OC-Feature-Vector AVP sent as part of the OC-Supported-Features AVP included in answer messages for transaction where the request contained an OC-Supported-Features AVP. This is per the procedures defined in [RFC7683].

For peer reports the selected algorithm is reflected in the OC-Peer-Algo AVP sent as part of the OC-Supported-Features AVP included answer messages for transactions where the request contained an OC-Supported-Features AVP. This is per the procedures defined in [I-D.ietf-dime-agent-overload].

Editor's Node: The peer report specification is still under development and, as such, the above paragraph is subject to change.

5. Overload Report Handling

This section describes any changes to the behavior defined in [RFC7683] for handling of overload reports when the rate overload abatement algorithm is used.

5.1. Reporting Node Overload Control State

A reporting node that uses the rate abatement algorithm SHOULD maintain reporting node Overload Control State (OCS) for each reacting node to which it sends a rate Overload Report (OLR).

This is different from the behavior defined in [RFC7683] where a single loss percentage sent to all reacting nodes.

A reporting node SHOULD maintain OCS entries when using the rate abatement algorithm per supported Diameter application, per targeted reacting node and per report type.

A rate OCS entry is identified by the tuple of Application-Id, report type and DiameterIdentity of the target of the rate OLR.

A reporting node that supports the rate abatement algorithm MUST include the rate of its abatement algorithm in the OC-Maximum-Rate AVP when sending a rate OLR.

All other elements for the OCS defined in [RFC7683] and [I-D.ietf-dime-agent-overload] also apply to the reporting nodes OCS when using the rate abatement algorithm.

5.2. Reacting Node Overload Control State

A reacting node that supports the rate abatement algorithm MUST indicate rate as the selected abatement algorithm in the reacting node OCS when receiving a rate OLR.

A reacting node that supports the rate abatement algorithm MUST include the rate specified in the OC-Maximum-Rate AVP included in the

OC-OLR AVP as an element of the abatement algorithm specific portion of reacting node OCS entries.

All other elements for the OCS defined in [RFC7683] and [I-D.ietf-dime-agent-overload] also apply to the reporting nodes OCS when using the rate abatement algorithm.

5.3. Reporting Node Maintenance of Overload Control State

A reporting node that has selected the rate overload abatement algorithm and enters an overload condition MUST indicate rate as the abatement algorithm in the resulting reporting node OCS entries.

A reporting node that has selected the rate abatement algorithm and enters an overload condition MUST indicate the selected rate in the resulting reporting node OCS entries.

When selecting the rate algorithm in the response to a request that contained an OC-Supporting-Features AVP with an OC-Feature-Vector AVP indicating support for the rate feature, a reporting node MUST ensure that a reporting node OCS entry exists for the target of the overload report. The target is defined as follows:

- o For Host reports the target is the DiameterIdentity contained in the Origin-Host AVP received in the request.
- o For Realm reports the target is the DiameterIdentity contained in the Origin-Realm AVP received in the request.
- o For Peer reports the target is the DiameterIdentity of the Diameter Peer from which the request was received.

5.4. Reacting Node Maintenance of Overload Control State

When receiving an answer message indicating that the reporting node has selected the rate algorithm, a reacting node MUST indicate the rate abatement algorithm in the reacting node OCS entry for the reporting node.

A reacting node receiving an overload report for the rate abatement algorithm MUST save the rate received in the OC-Maximum-Rate AVP contained in the OC-OLR AVP in the reacting node OCS entry.

5.5. Reporting Node Behavior for Rate Abatement Algorithm

When in an overload condition with rate selected as the overload abatement algorithm and when handling a request that contained an OC-Supported-Features AVP that indicated support for the rate abatement

algorithm, a reporting node SHOULD include an OC-OLR AVP for the rate algorithm using the parameters stored in the reporting node OCS for the target of the overload report.

When sending an overload report for the rate algorithm, the OC-Maximum-Rate AVP MUST be included and the OC-Reduction-Percentage AVP MUST NOT be included.

5.6. Reacting Node Behavior for Rate Abatement Algorithm

When determining if abatement treatment should be applied to a request being sent to a reporting node that has selected the rate overload abatement algorithm, the reacting node MAY use the algorithm detailed in Section 7.

Note: Other algorithms for controlling the rate can be implemented by the reacting node as long as they result in the correct rate of traffic being sent to the reporting node.

Once a determination is made by the reacting node that an individual Diameter request is to be subjected to abatement treatment then the procedures for throttling and diversion defined in [RFC7683] and [I-D.ietf-dime-agent-overload] apply.

6. Rate Abatement Algorithm AVPs

6.1. OC-Supported-Features AVP

The rate algorithm does not add any new AVPs to the OC-Supported-Features AVP.

The rate algorithm does add a new feature bit to be carried in the OC-Feature-Vector AVP.

6.1.1. OC-Feature-Vector AVP

This extension adds the following capabilities to the OC-Feature-Vector AVP.

OLR_RATE_ALGORITHM (0x0000000000000004)

When this flag is set by the overload control endpoint it indicates that the DOIC Node supports the rate overload control algorithm.

6.2. OC-OLR AVP

This extension defines the OC-Maximum-Rate AVP to be an optional part of the OC-OLR AVP.

```

OC-OLR ::= < AVP Header: TBD2 >
          < OC-Sequence-Number >
          < OC-Report-Type >
          [ OC-Reduction-Percentage ]
          [ OC-Validity-Duration ]
          [ SourceID ]
          [ OC-Maximum-Rate ]
          * [ AVP ]
    
```

This extension makes no changes to the other AVPs that are part of the OC-OLR AVP.

This extension does not define new overload report types. The existing report types of host and realm defined in [RFC7683] apply to the rate control algorithm. The peer report type defined in [I-D.ietf-dime-agent-overload] also applies to the rate control algorithm.

6.2.1. OC-Maximum-Rate AVP

The OC-Maximum-Rate AVP (AVP code TBD1) is of type Unsigned32 and describes the maximum rate that the sender is requested to send traffic. This is specified in terms of requests per second.

A value of zero indicates that no traffic is to be sent.

6.3. Attribute Value Pair flag rules

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules	
				MUST	NOT
OC-Maximum-Rate	TBD1	6.2	Unsigned32	V	

7. Rate Based Abatement Algorithm

This section is pulled from [RFC7415], with minor changes needed to make it apply to the Diameter protocol.

7.1. Overview

The reporting node is the one protected by the overload control algorithm defined here. The reacting node is the one that abates traffic towards the server.

Following the procedures defined in [draft-ietf-dime-doic], the reacting node and reporting node signal one another support for rate-based overload control.

Then periodically, the reporting node relies on internal measurements (e.g. CPU utilization or queuing delay) to evaluate its overload state and estimate a target maximum Diameter request rate in number of requests per second (as opposed to target percent reduction in the case of loss-based abatement).

When in an overloaded state, the reporting node uses the OC-OLR AVP to inform reacting nodes of its overload state and of the target Diameter request rate.

Upon receiving the overload report with a target maximum Diameter request rate, each reacting node applies abatement treatment for new Diameter requests towards the reporting node.

7.2. Reporting Node Behavior

The actual algorithm used by the reporting node to determine its overload state and estimate a target maximum Diameter request rate is beyond the scope of this document.

However, the reporting node MUST periodically evaluate its overload state and estimate a target Diameter request rate beyond which it would become overloaded. The reporting node must allocate a portion of the target Diameter request rate to each of its reacting nodes. The reporting node may set the same rate for every reacting node, or may set different rates for different reacting node.

The maximum rate determined by the reporting node for a reacting node applies to the entire stream of Diameter requests, even though abatement may only affect a particular subset of the requests, since the reacting node might apply priority as part of its decision of which requests to abate.

When setting the maximum rate for a particular reacting node, the reporting node may need take into account the workload (e.g. CPU load per request) of the distribution of message types from that reacting node. Furthermore, because the reacting node may prioritize the specific types of messages it sends while under overload restriction, this distribution of message types may be different from the message distribution for that reacting node under non-overload conditions (e.g., either higher or lower CPU load).

Note that the AVP for the rate algorithm is an upper bound (in request messages per second) on the traffic sent by the reacting node to the reporting node. The reacting node may send traffic at a rate significantly lower than the upper bound, for a variety of reasons.

In other words, when multiple reacting nodes are being controlled by an overloaded reporting node, at any given time some reacting nodes may receive requests at a rate below its target maximum Diameter request rate while others above that target rate. But the resulting request rate presented to the overloaded reporting node will converge towards the target Diameter request rate.

Upon detection of overload, and the determination to invoke overload controls, the reporting node MUST follow the specifications in [RFC7683] to notify its clients of the allocated target maximum Diameter request rate and to notify them that the rate overload abatement is in effect.

The reporting node MUST use the OC-Maximum-Rate AVP defined in this specification to communicate a target maximum Diameter request rate to each of its clients.

7.3. Reacting Node Behavior

7.3.1. Default Algorithm

In determining whether or not to transmit a specific message, the reacting node can use any algorithm that limits the message rate to the OC-Maximum-Rate AVP value in units of messages per second. For ease of discussion, we define $T = 1/[\text{OC-Maximum-Rate}]$ as the target inter-Diameter request interval. It may be strictly deterministic, or it may be probabilistic. It may, or may not, have a tolerance factor, to allow for short bursts, as long as the long term rate remains below $1/T$.

The algorithm may have provisions for prioritizing traffic.

If the algorithm requires other parameters (in addition to "T", which is $1/\text{OC-Maximum-Rate}$), they may be set autonomously by the reacting

node, or they may be negotiated independently between reacting node and reporting node.

In either case, the coordination is out of scope for this document. The default algorithms presented here (one with and one without provisions for prioritizing traffic) are only examples.

To apply abatement treatment to new Diameter requests at the rate specified in the OC-Maximum-Rate AVP value sent by the reporting node to its reacting nodes, the reacting node MAY use the proposed default algorithm for rate-based control or any other equivalent algorithm that forward messages in conformance with the upper bound of $1/T$ messages per second.

The default Leaky Bucket algorithm presented here is based on [ITU-T Rec. I.371] Appendix A.2. The algorithm makes it possible for reacting nodes to deliver Diameter requests at a rate specified in the OC-Maximum-Rate value with tolerance parameter TAU (preferably configurable).

Conceptually, the Leaky Bucket algorithm can be viewed as a finite capacity bucket whose real-valued content drains out at a continuous rate of 1 unit of content per time unit and whose content increases by the increment T for each forwarded Diameter request. T is computed as the inverse of the rate specified in the OC-Maximum-Rate AVP value, namely $T = 1 / \text{OC-Maximum-Rate}$.

Note that when the OC-Maximum-Rate value is 0 with a non-zero OC-Validity-Duration, then the reacting node should apply abatement treatment to 100% of Diameter requests destined to the overloaded reporting node. However, when the OC-Validity-Duration value is 0, the reacting node should stop applying abatement treatment.

If, at a new Diameter request arrival, the content of the bucket is less than or equal to the limit value TAU, then the Diameter request is forwarded to the server; otherwise, the abatement treatment is applied to the Diameter request.

Note that the capacity of the bucket (the upper bound of the counter) is $(T + \text{TAU})$.

The tolerance parameter TAU determines how close the long-term admitted rate is to an ideal control that would admit all Diameter requests for arrival rates less than $1/T$ and then admit Diameter requests precisely at the rate of $1/T$ for arrival rates above $1/T$. In particular at mean arrival rates close to $1/T$, it determines the tolerance to deviation of the inter-arrival time from T (the larger

TAU the more tolerance to deviations from the inter-departure interval T).

This deviation from the inter-departure interval influences the admitted rate burstyness, or the number of consecutive Diameter requests forwarded to the reporting node (burst size proportional to TAU over the difference between $1/T$ and the arrival rate).

In situations where reacting nodes are configured with some knowledge about the reporting node (e.g., operator pre-provisioning), it can be beneficial to choose a value of TAU based on how many reacting nodes will be sending requests to the reporting node.

Reporting nodes with a very large number of reacting nodes, each with a relatively small arrival rate, will generally benefit from a smaller value for TAU in order to limit queuing (and hence response times) at the reporting node when subjected to a sudden surge of traffic from all reacting nodes. Conversely, a reporting node with a relatively small number of reacting nodes, each with proportionally larger arrival rate, will benefit from a larger value of TAU.

Once the control has been activated, at the arrival time of the k-th new Diameter request, $ta(k)$, the content of the bucket is provisionally updated to the value

$$X' = X - (ta(k) - LCT)$$

where X is the value of the leaky bucket counter after arrival of the last forwarded Diameter request, and LCT is the time at which the last Diameter request was forwarded.

If X' is less than or equal to the limit value TAU, then the new Diameter request is forwarded and the leaky bucket counter X is set to X' (or to 0 if X' is negative) plus the increment T, and LCT is set to the current time $ta(k)$. If X' is greater than the limit value TAU, then the abatement treatment is applied to the new Diameter request and the values of X and LCT are unchanged.

When the first response from the reporting node has been received indicating control activation ($OC-Validity-Duration > 0$), LCT is set to the time of activation, and the leaky bucket counter is initialized to the parameter TAU0 (preferably configurable) which is 0 or larger but less than or equal to TAU.

TAU can assume any positive real number value and is not necessarily bounded by T.

$TAU=4*T$ is a reasonable compromise between burst size and abatement rate adaptation at low offered rate.

Note that specification of a value for TAU, and any communication or coordination between servers, is beyond the scope of this document.

A reference algorithm is shown below.

No priority case:

```
// T: inter-transmission interval, set to 1 / OC-Maximum-Rate
// TAU: tolerance parameter
// ta: arrival time of the most recent arrival
// LCT: arrival time of last SIP request that was sent to the server
//      (initialized to the first arrival time)
// X: current value of the leaky bucket counter (initialized to
//     TAU0)

// After most recent arrival, calculate auxiliary variable Xp
Xp = X - (ta - LCT);

if (Xp <= TAU) {
    // Transmit SIP request
    // Update X and LCT
    X = max (0, Xp) + T;
    LCT = ta;
} else {
    // Reject SIP request
    // Do not update X and LCT
}
```

7.3.2. Priority Treatment

The reacting node is responsible for applying message priority and for maintaining two categories of requests: Request candidates for reduction, requests not subject to reduction (except under extenuating circumstances when there aren't any messages in the first category that can be reduced).

Accordingly, the proposed Leaky bucket implementation is modified to support priority using two thresholds for Diameter requests in the set of request candidates for reduction. With two priorities, the proposed Leaky bucket requires two thresholds $TAU1 < TAU2$:

- o All new requests would be admitted when the leaky bucket counter is at or below TAU1,

- o Only higher priority requests would be admitted when the leaky bucket counter is between TAU1 and TAU2,
- o All requests would be rejected when the bucket counter is above TAU2.

This can be generalized to n priorities using n thresholds for $n > 2$ in the obvious way.

With a priority scheme that relies on two tolerance parameters (TAU2 influences the priority traffic, TAU1 influences the non-priority traffic), always set $TAU1 \leq TAU2$ (TAU is replaced by TAU1 and TAU2). Setting both tolerance parameters to the same value is equivalent to having no priority. TAU1 influences the admitted rate the same way as TAU does when no priority is set. And the larger the difference between TAU1 and TAU2, the closer the control is to strict priority queuing.

TAU1 and TAU2 can assume any positive real number value and is not necessarily bounded by T.

Reasonable values for TAU0, TAU1 & TAU2 are:

- o $TAU0 = 0$,
- o $TAU1 = 1/2 * TAU2$, and
- o $TAU2 = 10 * T$.

Note that specification of a value for TAU1 and TAU2, and any communication or coordination between servers, is beyond the scope of this document.

A reference algorithm is shown below.

Priority case:

```
// T: inter-transmission interval, set to 1 / OC-Maximum-Rate
// TAU1: tolerance parameter of no priority Diameter requests
// TAU2: tolerance parameter of priority Diameter requests
// ta: arrival time of the most recent arrival
// LCT: arrival time of last Diameter request that was sent to the server
//      (initialized to the first arrival time)
// X: current value of the leaky bucket counter (initialized to
//    TAU0)

// After most recent arrival, calculate auxiliary variable Xp
Xp = X - (ta - LCT);

if (AnyRequestReceived && Xp <= TAU1) || (PriorityRequestReceived &&
Xp <= TAU2 && Xp > TAU1) {
  // Transmit Diameter request
  // Update X and LCT
  X = max (0, Xp) + T;
  LCT = ta;
} else {
  // Apply abatement treatment to Diameter request
  // Do not update X and LCT
}
```

7.3.3. Optional Enhancement: Avoidance of Resonance

As the number of reacting node sources of traffic increases and the throughput of the reporting node decreases, the maximum rate admitted by each reacting node needs to decrease, and therefore the value of T becomes larger. Under some circumstances, e.g. if the traffic arises very quickly simultaneously at many sources, the occupancies of each bucket can become synchronized, resulting in the admissions from each source being close in time and batched or very 'peaky' arrivals at the reporting node, which not only gives rise to control instability, but also very poor delays and even lost messages. An appropriate term for this is 'resonance' [Erramilli].

If the network topology is such that resonance can occur, then a simple way to avoid resonance is to randomize the bucket occupancy at two appropriate points -- at the activation of control and whenever the bucket empties -- as described below.

After updating the value of the leaky bucket to X' , generate a value u as follows:

```
if  $X' > 0$ , then  $u=0$ 
```

```
else if  $X' <= 0$ , then let  $u$  be set to a random value uniformly
distributed between  $-1/2$  and  $+1/2$ 
```

Then (only) if the arrival is admitted, increase the bucket by an amount $T + uT$, which will therefore be just T if the bucket hadn't emptied, or lie between $T/2$ and $3T/2$ if it had.

This randomization should also be done when control is activated, i.e. instead of simply initializing the leaky bucket counter to $TAU0$, initialize it to $TAU0 + uT$, where u is uniformly distributed as above. Since activation would have been a result of response to a request sent by the reacting node, the second term in this expression can be interpreted as being the bucket increment following that admission.

This method has the following characteristics:

- o If $TAU0$ is chosen to be equal to TAU and all sources activate control at the same time due to an extremely high request rate, then the time until the first request admitted by each reacting node would be uniformly distributed over $[0, T]$;
- o The maximum occupancy is $TAU + (3/2)T$, rather than $TAU + T$ without randomization;
- o For the special case of 'classic gapping' where $TAU=0$, then the minimum time between admissions is uniformly distributed over $[T/2, 3T/2]$, and the mean time between admissions is the same, i.e. $T+1/R$ where R is the request arrival rate.
- o At high load randomization rarely occurs, so there is no loss of precision of the admitted rate, even though the randomized 'phasing' of the buckets remains.

8. IANA Consideration

8.1. AVP codes

New AVPs defined by this specification are listed in Section 6. All AVP codes are allocated from the 'Authentication, Authorization, and Accounting (AAA) Parameters' AVP Codes registry.

8.2. New registries

There are no new IANA registries introduced by this document.

9. Security Considerations

The rate overload abatement mechanism is an extension to the base Diameter overload mechanism. As such, all of the security

considerations outlined in [RFC7683] apply to the rate overload abatement mechanism.

10. Acknowledgements

11. References

11.1. Normative References

- [I-D.ietf-dime-agent-overload]
Donovan, S., "Diameter Agent Overload", draft-ietf-dime-agent-overload-00 (work in progress), December 2014.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.
- [RFC7683] Korhonen, J., Ed., Donovan, S., Ed., Campbell, B., and L. Morand, "Diameter Overload Indication Conveyance", RFC 7683, DOI 10.17487/RFC7683, October 2015, <<http://www.rfc-editor.org/info/rfc7683>>.

11.2. Informative References

- [Erramilli]
Erramilli, A. and L. Forys, "Traffic Synchronization Effects In Teletraffic Systems", 1991.
- [RFC7415] Noel, E. and P. Williams, "Session Initiation Protocol (SIP) Rate Control", RFC 7415, DOI 10.17487/RFC7415, February 2015, <<http://www.rfc-editor.org/info/rfc7415>>.

Authors' Addresses

Steve Donovan (editor)
Oracle
17210 Campbell Road
Dallas, Texas 75254
United States

Email: srdonovan@usdonovans.com

Eric Noel
AT&T Labs
200s Laurel Avenue
Middletown, NJ 07747
United States

Email: ecnoel@research.att.com

Diameter Maintenance and Extensions (DIME)
Internet-Draft
Intended status: Standards Track
Expires: September 14, 2017

M. Jones
M. Liebsch
L. Morand
March 13, 2017

Diameter Group Signaling
draft-ietf-dime-group-signaling-08.txt

Abstract

In large network deployments, a single Diameter node can support over a million concurrent Diameter sessions. Recent use cases have revealed the need for Diameter nodes to apply the same operation to a large group of Diameter sessions concurrently. The Diameter base protocol commands operate on a single session so these use cases could result in many thousands of command exchanges to enforce the same operation on each session in the group. In order to reduce signaling, it would be desirable to enable bulk operations on all (or part of) the sessions managed by a Diameter node using a single or a few command exchanges. This document specifies the Diameter protocol extensions to achieve this signaling optimization.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 14, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Protocol Overview	4
3.1.	Building and Modifying Session Groups	4
3.2.	Issuing Group Commands	4
3.3.	Permission Considerations	4
4.	Protocol Description	6
4.1.	Session Grouping Capability Discovery	7
4.1.1.	Explicit Capability Discovery	7
4.1.2.	Implicit Capability Discovery	7
4.2.	Session Grouping	7
4.2.1.	Group assignment at session initiation	8
4.2.2.	Removing a session from a session group	11
4.2.3.	Mid-session group assignment modifications	12
4.3.	Deleting a Session Group	13
4.4.	Performing Group Operations	13
4.4.1.	Sending Group Commands	13
4.4.2.	Receiving Group Commands	14
4.4.3.	Error Handling for Group Commands	14
4.4.4.	Single-Session Fallback	15
5.	Operation with Proxy Agents	15
6.	Commands Formatting	16
6.1.	Formatting Example: Group Re-Auth-Request	16
7.	Attribute-Value-Pairs (AVP)	17
7.1.	Session-Group-Info AVP	17
7.2.	Session-Group-Control-Vector AVP	18
7.3.	Session-Group-Id AVP	18
7.4.	Group-Response-Action AVP	19
7.5.	Session-Group-Capability-Vector AVP	19
8.	Result-Code AVP Values	19
9.	IANA Considerations	19
9.1.	AVP Codes	20
10.	Security Considerations	20
11.	Acknowledgments	21
12.	Normative References	21
	Appendix A. Session Management -- Exemplary Session State	

Machine	21
A.1. Use of groups for the Authorization Session State Machine	21
Authors' Addresses	26

1. Introduction

In large network deployments, a single Diameter node can support over a million concurrent Diameter sessions. Recent use cases have revealed the need for Diameter nodes to apply the same operation to a large group of Diameter sessions concurrently. For example, a policy decision point may need to modify the authorized quality of service for all active users having the same type of subscription. The Diameter base protocol commands operate on a single session so these use cases could result in many thousands of command exchanges to enforce the same operation on each session in the group. In order to reduce signaling, it would be desirable to enable bulk operations on all (or part of) the sessions managed by a Diameter node using a single or a few command exchanges.

This document describes mechanisms for grouping Diameter sessions and applying Diameter commands, such as performing re-authentication, re-authorization, termination and abortion of sessions to a group of sessions. This document does not define a new Diameter application. Instead it defines mechanisms, commands and AVPs that may be used by any Diameter application that requires management of groups of sessions.

These mechanisms take the following design goals and features into account:

- o Minimal impact to existing applications
- o Extension of existing commands' Command Code Format (CCF) with optional AVPs to enable grouping and group operations
- o Fallback to single session operation
- o Implicit discovery of capability to support grouping and group operations in case no external mechanism is available to discover a Diameter peer's capability to support session grouping and session group operations

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document uses terminology defined in [RFC6733].

3. Protocol Overview

3.1. Building and Modifying Session Groups

Client and Server can assign a new Diameter session to a group, e.g. in case the subscription profile of the associated user has similar characteristics as the profile of other users whose Diameter session has been assigned to one or multiple groups. A single command can be issued and applied to all sessions associated with such group(s), e.g. to adjust common profile or policy settings.

The assignment of a Diameter session to a group can be changed mid-session. For example, if a user's subscription profile changes mid-session, a Diameter server may remove the session from its current group and assign the session to a different group that is more appropriate for the new subscription profile.

In case of mobile users, the user's session may get transferred to a new Diameter client during handover and assigned to a different group, which is maintained at the new Diameter client, mid-session.

A session group, which has sessions assigned, can be deleted, e.g. due to a change in multiple users' subscription profile so that the group's assigned sessions do not share certain characteristics anymore. Deletion of such group requires subsequent individual treatment of each of the assigned sessions. A node may decide to assign some of these sessions to any other existing or new group.

3.2. Issuing Group Commands

Changes in the network condition may result in the Diameter server's decision to close all sessions in a given group. The server issues a single Session Termination Request (STR) command, identifying the group of sessions which are to be terminated. The Diameter client treats the STR as group command and initiates termination of all sessions associated with the identified group. Subsequently, the client confirms successful termination of these sessions to the server by sending a single Session Termination Answer (STA) command, which includes the identifier of the group.

3.3. Permission Considerations

Permission considerations in the context of this draft apply to the permission of Diameter nodes to build new session groups, to assign/remove a session to/from a session group and to delete an existing session group.

This specification follows the most flexible model where both, a Diameter client and a Diameter server can create a new group and assign a new identifier to that session group. When a Diameter node decides to create a new session group, e.g. to group all sessions which share certain characteristics, the node builds a session group identifier according to the rules described in Section 7.3 and becomes the owner of the group. This specification does not constrain the permission to add or remove a session to/from a session group to the group owner, instead each node can add a session to any known group or remove a session from a group. A session group is deleted and its identifier released after the last session has been removed from the session group. Also the modification of groups in terms of moving a session from one session group to a different session group is permitted to any Diameter node. A Diameter node can delete a session group and its group identifier mid-session, resulting in individual treatment of the sessions which have been previously assigned to the deleted group. Prerequisite for deletion of a session group is that the Diameter node created the session beforehand, hence the node became the group owner.

The enforcement of more constrained permissions is left to the specification of a particular group signaling enabled Diameter application and compliant implementations of such application MUST enforce the associated permission model. Details about enforcing a more constraint permission model are out of scope of this specification. For example, a more constrained model could require that a client MUST NOT remove a session from a group which is owned by the server.

The following table depicts the permission considerations as per the present specification:

Operation	Server	Client
Create a new Session Group (Diameter node becomes the group owner)	X	X
Assign a Session to an owned Session Group	X	X
Assign a Session to a non-owned Session Group	X	X
Remove a Session from an owned Session Group	X	X
Remove a Session from a non-owned Session Group	X	X
Remove a Session from a Session Group where the Diameter node created the assignment	X	X
Remove a Session from a Session Group where a different Diameter node created the assignment		
Overrule a different Diameter node's group assignment *)		
Delete a Session Group which is owned by the Diameter node	X	X
Delete a Session Group which is not owned by the Diameter node		

Default Permission as per this Specification

*) Editors' note: The protocol specification in this document does not consider overruling a node's assignment of a session to a session group. Here, overruling is to be understood as a node changing the group(s) assignment as per the node's request. Group signaling enabled applications may take such protocol support and associated protocol semantics into account in their specification. One exception is adopted in this specification, which allows a Diameter server to reject a group assignment as per the client's request.

4. Protocol Description

4.1. Session Grouping Capability Discovery

Diameter nodes SHOULD assign a session to a session group and perform session group operations with a node only after having ensured that the node announced associated support beforehand.

4.1.1. Explicit Capability Discovery

New Diameter applications may consider support for Diameter session grouping and for performing group commands during the standardization process. Such applications provide intrinsic discovery for the support of group commands and announce this capability through the assigned application ID.

System- and deployment-specific means, as well as out-of-band mechanisms for capability exchange can be used to announce nodes' support for session grouping and session group operations. In such case, the optional Session-Group-Capability-Vector AVP, as described in Section 4.1.2 can be omitted in Diameter messages being exchanged between nodes.

4.1.2. Implicit Capability Discovery

If no explicit mechanism for capability discovery is deployed to enable Diameter nodes to learn about nodes' capability to support session grouping and group commands, a Diameter node SHOULD append the Session-Group-Capability-Vector AVP to any Diameter messages exchanged with its nodes to announce its capability to support session grouping and session group operations. Implementations following the specification as per this document set the `BASE_SESSION_GROUP_CAPABILITY` flag of the Session-Group-Capability-Vector AVP.

When a Diameter node receives at least one Session-Group-Capability-Vector AVP from a node with the `BASE_SESSION_GROUP_CAPABILITY` flag set, the Diameter node maintains a log to remember the node's capability to support group commands.

4.2. Session Grouping

This specification does not limit the number of session groups, to which a single session is assigned. It is left to the application to determine the policy of session grouping. In case an application facilitates a session to belong to multiple session groups, the application MUST maintain consistency of associated application session states for these multiple session groups.

Either Diameter node (client or server) can initiate the assignment of a session to a single or multiple session groups. Modification of a group by removing or adding a single or multiple user sessions can be initiated and performed mid-session by either Diameter node. Diameter AAA applications typically assign client and server roles to the Diameter nodes, which are referred to as relevant Diameter nodes to utilize session grouping and issue group commands. Section 5 describes particularities about session grouping and performing group commands when relay agents or proxies are deployed.

Diameter nodes, which are group-aware, MUST store and maintain an entry about the group assignment together with a session's state. A list of all known session groups should be locally maintained on each node, each group pointing to individual sessions being assigned to the group. A Diameter node MUST also keep a record about sessions, which have been assigned to a session group by itself.

4.2.1. Group assignment at session initiation

To assign a session to a group at session initiation, a Diameter client sends a service specific request, e.g. NASREQ AA-Request [RFC4005], containing one or more session group identifiers. Each of these groups MUST be identified by a unique Session-Group-Id contained in a separate Session-Group-Info AVP as specified in Section 7.

The client may choose one or multiple session groups from a list of existing session groups. Alternatively, the client may decide to create a new group to which the session is assigned and identify itself in the <DiameterIdentity> portion of the Session-Group-Id AVP as per Section 7.3

The client MUST set the SESSION_GROUP_ALLOCATION_ACTION flag of the Session-Group-Control-Vector AVP in each appended Session-Group-Info AVP to indicate that the session contained in the request should be assigned to the identified session group.

The client may also indicate in the request that the server is responsible for the assignment of the session in one or multiple sessions owned by the server. In such a case, the client MUST include the Session-Group-Info AVP in the request including the Session-Group-Control-Vector AVP with the SESSION_GROUP_ALLOCATION_ACTION flag set but no Session-Group-Id AVP.

If the Diameter server receives a command request from a Diameter client and the command comprises at least one Session-Group-Info AVP having the SESSION_GROUP_ALLOCATION_ACTION flag in the Session-Group-Control-Vector AVP set, the server can accept or reject the request

for group assignment. Reasons for rejection may be e.g. lack of resources for managing additional groups. When rejected, the session MUST NOT be assigned to any session group.

If the Diameter server accepts the client's request for a group assignment, the server MUST assign the new session to each of the one or multiple identified session groups when present in the Session-Group-Info AVP. In case one or multiple identified session groups are not already stored by the server, the server MUST store the new identified group(s) to its local list of known session groups. When sending the response to the client, e.g. a service-specific auth response as per NASREQ AA-Answer [RFC4005], the server MUST include all Session-Group-Info AVPs as received in the client's request.

In addition to the one or multiple session groups identified in the client's request, the server may decide to assign the new session to one or multiple additional groups. In such a case, the server MUST add to the response the additional Session-Group-Info AVPs, each identifying a session group to which the new session is assigned by the server. Each of the Session-Group-Info AVP added by the server MUST have the SESSION_GROUP_ALLOCATION_ACTION flag set in the Session-Group-Control-Vector AVP set.

If the Diameter server rejects the client's request for a group assignment, the server sends the response to the client, e.g. a service-specific auth response as per NASREQ AA-Answer [RFC4005], and MUST include all Session-Group-Info AVPs as received in the client's request (if any) while clearing the SESSION_GROUP_ALLOCATION_ACTION flag of the Session-Group-Control-Vector AVP. The server MAY accept the client's request for the identified session but refuse the session's assignment to any session group. The server sends the response to the client indicating success in the result code. In such case the session is treated as single session without assignment to any session group by the Diameter nodes.

If the Diameter server accepts the client's request for a group assignment, but the assignment of the session to one or some of the multiple identified session groups fails, the session group assignment is treated as failure. In such case the session is treated as single session without assignment to any session group by the Diameter nodes. The server sends the response to the client and MAY include as information to the client only those Session-Group-Info AVPs for which the group assignment failed. The SESSION_GROUP_ALLOCATION_ACTION flag of included Session-Group-Info AVPs MUST be cleared.

If the Diameter server receives a command request from a Diameter client and the command comprises one or multiple Session-Group-Info

AVPs and none of them includes a Session-Group-Id AVP, the server MAY decide to assign the session to one or multiple session groups. For each session group, to which the server assigns the new session, the server includes a Session-Group-Info AVP with the Session-Group-Id AVP identifying a session group in the response sent to the client. Each of the Session-Group-Info AVPs included by the server MUST have the SESSION_GROUP_ALLOCATION_ACTION flag of the Session-Group-Control-Vector AVP set.

If the Diameter server receives a command request from a Diameter client and the command does not contain any Session-Group-Info AVP, the server MUST NOT assign the new session to any session group but treat the request as for a single session. The server MUST NOT return any Session-Group-Info AVP in the command response.

If the Diameter client receives a response to its previously issued request from the server and the response comprises at least one Session-Group-Info AVP having the SESSION_GROUP_ALLOCATION_ACTION flag of the associated Session-Group-Control-Vector AVP set, the client MUST add the new session to all session groups as identified in the one or multiple Session-Group-Info AVPs. If the Diameter client fails to add the session to one or more session groups as identified in the one or multiple Session-Group-Info AVPs, the client MUST terminate the session. The client MAY send a subsequent request for session initiation to the server without requesting the assignment of the session to a session group

If the Diameter client receives a response to its previously issued request from the server and the one or more Session-Group-Info AVPs have the SESSION_GROUP_ALLOCATION_ACTION flag of the associated Session-Group-Control-Vector AVP cleared, the client MUST terminate the assignment of the session to the one or multiple groups. If the response from the server indicates success in the result code but solely the assignment of the session to a session group has been rejected by the server, the client treats the session as single session without group assignment.

A Diameter client, which sent a request for session initiation to a Diameter server and appended a single or multiple Session-Group-Id AVPs but cannot find any Session-Group-Info AVP in the associated response from the Diameter server proceeds as if the request was processed for a single session. When the Diameter client is confident that the Diameter server supports session grouping and group signaling, the Diameter client SHOULD NOT retry to request group assignment for this session, but MAY try to request group assignment for other new sessions.

4.2.2. Removing a session from a session group

When a Diameter client decides to remove a session from a particular session group, the client sends a service-specific re-authorization request to the server and adds one Session-Group-Info AVP to the request for each session group, from which the client wants to remove the session. The session, which is to be removed from a group, is identified in the Session-Id AVP of the command request. The `SESSION_GROUP_ALLOCATION_ACTION` flag of the Session-Group-Control-Vector AVP in each Session-Group-Info AVP **MUST** be cleared to indicate removal of the session from the session group identified in the associated Session-Group-id AVP.

When a Diameter client decides to remove a session from all session groups, to which the session has been previously assigned, the client sends a service-specific re-authorization request to the server and adds a single Session-Group-Info AVP to the request which has the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared and the Session-Group-Id AVP omitted. The session, which is to be removed from all groups, to which the session has been previously assigned, is identified in the Session-Id AVP of the command request.

If the Diameter server receives a request from the client which has at least one Session-Group-Info AVP appended with the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared, the server **MUST** remove the session from the session group identified in the associated Session-Group-Id AVP. If the request comprises at least one Session-Group-info AVP with the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared and no Session-Id AVP present, the server **MUST** remove the session from all session groups to which the session has been previously assigned. The server **MUST** include in its response to the requesting client all Session-Group-Id AVPs as received in the request.

When the Diameter server decides to remove a session from one or multiple particular session groups or from all session groups to which the session has been assigned beforehand, the server sends a Re-Authorization Request (RAR) or a service-specific server-initiated request to the client, indicating the session in the Session-Id AVP of the request. The client sends a Re-Authorization Answer (RAA) or a service-specific answer to respond to the server's request. The client subsequently sends service-specific re-authorization request containing one or multiple Session-Group-Info AVPs, each indicating a session group, to which the session had been previously assigned. To indicate removal of the indicated session from one or multiple session groups, the server sends a service-specific auth response to the client, containing a list of Session-Group-Info AVPs with the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared and the Session-Group-Id AVP identifying the session group, from which the session should be

removed. The server MAY include to the service-specific auth response a list of Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP identifying session groups to which the session remains subscribed. In case the server decides to remove the identified session from all session groups, to which the session has been previously assigned, the server includes in the service-specific auth response at least one Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and Session-Group-Id AVP absent.

4.2.3. Mid-session group assignment modifications

Either Diameter node (client or server) can modify the group membership of an active Diameter session according to the specified permission considerations.

To update an assigned group mid-session, a Diameter client sends a service-specific re-authorization request to the server, containing one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP present, identifying the session group to which the session should be assigned. With the same message, the client may send one or multiple Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and the Session-Group-Id AVP identifying the session group from which the identified session is to be removed. To remove the session from all previously assigned session groups, the client includes at least one Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and no Session-Group-Id AVP present. When the server received the service-specific re-authorization request, it MUST update its locally maintained view of the session groups for the identified session according to the appended Session-Group-Info AVPs. The server sends a service-specific auth response to the client containing one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP identifying the new session group to which the identified session has been assigned.

When a Diameter server enforces an update to the assigned groups mid-session, it sends a Re-Authorization Request (RAR) message to the client identifying the session, for which the session group lists are to be updated. The client responds with a Re-Authorization Answer (RAA) message. The client subsequently sends a service-specific re-authorization request containing one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP identifying the session group to which the session had been previously assigned. The server responds with a service-specific auth response and includes one or multiple Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag set and

the Session-Group-Id AVP identifying the session group, to which the identified session is to be assigned. With the same response message, the server may send one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and the Session-Group-Id AVP identifying the session groups from which the identified session is to be removed. When server wants to remove the session from all previously assigned session groups, it sends at least one Session-Group-Info AVP with the response having the SESSION_GROUP_ALLOCATION_ACTION flag cleared and no Session-Group-Id AVP present.

4.3. Deleting a Session Group

To delete a session group and release the associated Session-Group-Id value, the owner of a session group appends a single Session-Group-Info AVP having the SESSION_GROUP_STATUS_IND flag cleared and the Session-Group-Id AVP identifying the session group, which is to be deleted. The SESSION_GROUP_ALLOCATION_ACTION flag of the associated Session-Group-Control-Vector AVP MUST be cleared.

4.4. Performing Group Operations

4.4.1. Sending Group Commands

Either Diameter node (client or server) can request the recipient of a request to process an associated command for all sessions being assigned to one or multiple groups by identifying these groups in the request. The sender of the request appends for each group, to which the command applies, a Session-Group-Info AVP including the Session-Group-Id AVP to identify the associated session group. Both, the SESSION_GROUP_ALLOCATION_ACTION flag as well as the SESSION_GROUP_STATUS_IND flag MUST be set.

If the CCF of the request mandates a Session-Id AVP, the Session-Id AVP MUST identify one of the single sessions which is assigned to at least one of the groups being identified in the appended Session-Group-Id AVPs.

The sender of the request MUST indicate to the receiver how follow up message exchanges should be performed by appending a single instance of the Group-Response-Action AVP. Even if the request includes multiple instances of a Session-Group-Info AVP, the request MUST NOT comprise more than a single instance of a Group-Response-Action AVP. If the sender wants the receiver to perform follow up exchanges with a single command for all impacted groups, the sender sets the value of the Group-Response-Action AVP to ALL_GROUPS (1). If follow up message exchanges should be performed on a per-group basis in case multiple groups are identified in the group command, the value of the

Group-Response-Action AVP is set to PER_GROUP (2). A value set to PER_SESSION (3) indicates to the receiver that all follow up exchanges should be performed using a single message for each impacted session.

If the sender sends a request including the Group-Response-Action AVP set to ALL_GROUPS (1) or PER_GROUP (2), it MUST expect some delay before receiving the corresponding answer(s) as the answer(s) will only be sent back when the request is processed for all the sessions or all the session of a session group. If the process of the request is delay-sensitive, the sender SHOULD NOT set the Group-Response-Action AVP to ALL_GROUPS (1) or PER_GROUP (2). If the answer can be sent before the complete process of the request for all the sessions or if the request timeout timer is high enough, the sender MAY set the Group-Response-Action AVP to ALL_GROUPS (1) or PER_GROUP (2).

If the sender wants the receiver of the request to process the associated command solely for a single session, the sender does not append any group identifier, but identifies the relevant session in the Session-Id AVP.

4.4.2. Receiving Group Commands

A Diameter node receiving a request to process a command for a group of sessions, identifies the relevant groups according to the appended Session-Group-Id AVP in the Session-Group-Info AVP and processes the group command according to the appended Group-Response-Action AVP . If the received request identifies multiple groups in multiple appended Session-Group-Id AVPs, the receiver SHOULD process the associated command for each of these groups. If a session has been assigned to more than one of the identified groups, the receiver MUST process the associated command only once per session.

4.4.3. Error Handling for Group Commands

When a Diameter node receives a request to process a command for one or more session groups and the result of processing the command is an error that applies to all sessions in the identified groups, an associated protocol error MUST be returned to the source of the request. In such case, the sender of the request MUST fall back to single-session processing and the session groups, which have been identified in the group command, MUST be deleted according to the procedure described in Section 4.3.

When a Diameter node receives a request to process a command for one or more session groups and the result of processing the command succeeds for some sessions identified in one or multiple session groups, but fails for one or more sessions, the Result-Code AVP in

the response message SHOULD indicate `DIAMETER_LIMITED_SUCCESS` as per Section 7.1.2 of [RFC6733].

In case of limited success, the sessions, for which the processing of the group command failed, MUST be identified using a Failed-AVP AVP as per Section 7.5 of [RFC6733]. The sender of the request MUST fall back to single-session operation for each of the identified sessions, for which the group command failed. In addition, each of these sessions MUST be removed from all session groups to which the group command applied. To remove sessions from a session group, the Diameter client performs the procedure described in Section 4.2.2.

4.4.4. Single-Session Fallback

Either Diameter node can fall back to single session operation by ignoring and omitting the optional group session-specific AVPs. Fallback to single-session operation is performed by processing the Diameter command solely for the session identified in the mandatory Session-Id AVP. In such case, the response to the group command MUST NOT identify any group but identify solely the single session for which the command has been processed.

5. Operation with Proxy Agents

In case of a present stateful Proxy Agent between a Diameter client and a Diameter server, this specification assumes that the Proxy Agent is aware of session groups and session group handling. The Proxy MUST update and maintain consistency of its local session states as per the result of the group commands which are operated between a Diameter client and a server. In such case, the Proxy Agent MUST act as a Diameter server in front of the Diameter client and MUST act as a Diameter client in front of the Diameter server. Therefore, the client and server behavior described in Section 4 applies respectively to the stateful Proxy Agent.

In case a stateful Proxy Agent manipulates session groups, it MUST maintain consistency of session groups between a client and a server. This applies to a deployment where the Proxy Agent utilizes session grouping and performs group operations with, for example, a Diameter server, whereas the Diameter client is not aware of session groups. In such case the Proxy Agent must reflect the states associated with the session groups as individual session operations towards the client and ensure the client has a consistent view of each session. The same applies to a deployment where all nodes, the Diameter client and server, as well as the Proxy Agent are group-aware but the Proxy Agent manipulates groups, e.g. to adopt different administrative policies that apply to the client's domain and the server's domain.

Stateless Proxy Agents do not maintain any session state (only transaction state are maintained). Consequently, the notion of session group is transparent for any stateless Proxy Agent present between a Diameter client and a Diameter server handling session groups. Session group related AVPs being defined as optional AVP SHOULD be ignored by stateless Proxy Agents and SHOULD NOT be removed from the Diameter commands. If they are removed by the Proxy Agent for any reason, the Diameter client and Diameter server will discover the absence the related session group AVPs and will fall back to single-session processing, as described in Section 4.

6. Commands Formatting

This document does not specify new Diameter commands to enable group operations, but relies on command extensibility capability provided by the Diameter Base protocol. This section provides the guidelines to extend the CCF of existing Diameter commands with optional AVPs to enable the recipient of the command applying the command to all sessions associated with the identified group(s).

6.1. Formatting Example: Group Re-Auth-Request

A request for re-authentication of one or more groups of users is issued by appending one or multiple Session-Group-Id AVP(s), as well as a single instance of a Group-Response-Action AVP to the Re-Auth-Request (RAR). The one or multiple Session-Group-Id AVP(s) identify the associated group(s) for which the group re-authentication has been requested. The Group-Response-Action AVP identifies the expected means to perform and respond to the group command. The recipient of the group command initiates re-authentication for all users associated with the identified group(s). Furthermore, the sender of the group re-authentication request appends a Group-Response-Action AVP to provide more information to the receiver of the command about how to accomplish the group operation.

The value of the mandatory Session-Id AVP MUST identify a session associated with a single user, which is assigned to at least one of the groups being identified in the appended Session-Group-Id AVPs.

```

<RAR> ::= < Diameter Header: 258, REQ, PXY >
        < Session-Id >
        { Origin-Host }
        { Origin-Realm }
        { Destination-Realm }
        { Destination-Host }
        { Auth-Application-Id }
        { Re-Auth-Request-Type }
        [ User-Name ]
        [ Origin-State-Id ]
        * [ Proxy-Info ]
        * [ Route-Record ]
        [ Session-Group-Capability-Vector ]
        * [ Session-Group-Info ]
        [ Group-Response-Action ]
        * [ AVP ]
    
```

7. Attribute-Value-Pairs (AVP)

Attribute Name	AVP Code	Value Type	AVP Flag rules			
			MUST	MAY	SHOULD NOT	MUST NOT
Session-Group-Info	TBD1	Grouped		P		V
Session-Group-Control-Vector	TBD2	Unsigned32		P		V
Session-Group-Id	TBD3	OctetString		P		V
Group-Response-Action	TBD4	Unsigned32		P		V
Session-Group-Capability-Vector	TBD5	Unsigned32		P		V

AVPs for the Diameter Group Signaling

7.1. Session-Group-Info AVP

The Session-Group-Info AVP (AVP Code TBD1) is of type Grouped. It contains the identifier of the session group as well as an indication of the node responsible for session group identifier assignment.

```

Session-Group-Info ::= < AVP Header: TBD1 >
                        < Session-Group-Control-Vector >
                        [ Session-Group-Id ]
                        * [ AVP ]
    
```

7.2. Session-Group-Control-Vector AVP

The Session-Group-Control-Vector AVP (AVP Code TBD2) is of type Unsigned32 and contains a 32-bit flags field to control the group assignment at session-group aware nodes.

The following capabilities are defined in this document:

`SESSION_GROUP_ALLOCATION_ACTION (0x00000001)`

This flag indicates the action to be performed for the identified session. When this flag is set, it indicates that the identified Diameter session is to be assigned to the session group as identified by the Session-Group-Id AVP or the session's assignment to the session group identified in the Session-Group-Id AVP is still valid. When the flag is cleared, the identified Diameter session is to be removed from at least one session group. When the flag is cleared and the Session-Group-Info AVP identifies a particular session group in the associated Session-Group-Id AVP, the session is to be removed solely from the identified session group. When the flag is cleared and the Session-Group-Info AVP does not identify a particular session group (Session-Group-Id AVP is absent), the identified Diameter session is to be removed from all session groups, to which it has been previously assigned.

`SESSION_GROUP_STATUS_IND (0x00000010)`

This flag indicates the status of the session group identified in the associated Session-Group-Id AVP. The flag is set when the identified session group has just been created or is still active. If the flag is cleared, the identified session group is deleted and the associated Session-Group-Id is released. If the Session-Group-Info AVP does not comprise a Session-Group-Id AVP, this flag is meaningless and MUST be ignored by the receiver.

7.3. Session-Group-Id AVP

The Session-Group-Id AVP (AVP Code TBD3) is of type UTF8String and identifies a group of Diameter sessions.

The Session-Group-Id MUST be globally and eternally unique, as it is meant to uniquely identify a group of Diameter sessions without reference to any other information.

The default format of the Session-Group-id MUST comply to the format recommended for a Session-Id, as defined in the section 8.8 of the [RFC6733]. The <DiameterIdentity> portion of the Session-Group-Id MUST identify the Diameter node, which owns the session group.

7.4. Group-Response-Action AVP

The Group-Response-Action AVP (AVP Code TBD4) is of type Unsigned32 and contains a 32-bit address space representing values indicating how the node SHOULD issue follow up exchanges in response to a command which impacts multiple sessions. The following values are defined by this application:

ALL_GROUPS (1)

Follow up exchanges should be performed with a single message exchange for all impacted groups.

PER_GROUP (2)

Follow up exchanges should be performed with a message exchange for each impacted group.

PER_SESSION (3)

Follow up exchanges should be performed with a message exchange for each impacted session.

7.5. Session-Group-Capability-Vector AVP

The Session-Group-Capability-Vector AVP (AVP Code TBD5) is of type Unsigned32 and contains a 32-bit flags field to indicate capabilities in the context of session-group assignment and group operations.

The following capabilities are defined in this document:

BASE_SESSION_GROUP_CAPABILITY (0x00000001)

This flag indicates the capability to support session grouping and session group operations according to this specification.

8. Result-Code AVP Values

This document does not define new Result-Code [RFC6733] values for existing applications, which are extended to support group commands. Specification documents of new applications, which will have intrinsic support for group commands, may specify new Result-Codes.

9. IANA Considerations

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

9.1. AVP Codes

This specification requires IANA to register the following new AVPs from the AVP Code namespace defined in [RFC6733].

- o Session-Group-Info
- o Session-Group-Control-Vector
- o Session-Group-Id
- o Group-Response-Action
- o Session-Group-Capability-Vector

The AVPs are defined in Section 7.

10. Security Considerations

The security considerations of the Diameter protocol itself are discussed in [RFC6733]. Use of the AVPs defined in this document MUST take into consideration the security issues and requirements of the Diameter base protocol. In particular, the Session-Group-Info AVP (including the Session-group-Control-Vector and the Session-Group-Id AVPs) should be considered as a security-sensitive AVPs in the same manner than the Session-Id AVP in the Diameter base protocol [RFC6733].

The management of session groups relies upon the existing trust relationship between the Diameter client and the Diameter server managing the groups of sessions. This document defines a mechanism that allows a client or a server to act on multiple sessions at the same time using only one command. If the Diameter client or server is compromised, an attacker could launch DoS attacks by terminating a large number of sessions with a limited set of commands using the session group management concept.

According to the Diameter base protocol [RFC6733], transport connections between Diameter peers are protected by TLS/TCP, DTLS/SCTP or alternative security mechanisms that are independent of Diameter, such as IPsec. However, the lack of end-to-end security features makes it difficult to establish trust in the session group related information received from non-adjacent nodes. Any Diameter agent in the message path can potentially modify the content of the message and therefore the information sent by the Diameter client or the server. The DIME working group is currently working on solutions for providing end-to-end security features. When available, these features should enable the establishment of trust relationship

between non-adjacent nodes and the security required for session group management would normally rely on this end-to-end security. However, there is no assumption in this document that such end-to-end security mechanism will be available. It is only assume that the solution defined on this document relies on the security framework provided by the Diameter based protocol.

In some cases, a Diameter Proxy agent can act on behalf of a client or server. In such a case, the security requirements that normally apply to a client (or a server) apply equally to the Proxy agent.

11. Acknowledgments

The authors of this document want to thank Ben Campbell and Eric McMurry for their valuable comments to early versions of this draft. Furthermore, authors thank Steve Donovan and Mark Bales for the thorough review and comments on advanced versions of the WG document, which helped a lot to improve this specification.

12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4005] Calhoun, P., Zorn, G., Spence, D., and D. Mitton, "Diameter Network Access Server Application", RFC 4005, DOI 10.17487/RFC4005, August 2005, <<http://www.rfc-editor.org/info/rfc4005>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.

Appendix A. Session Management -- Exemplary Session State Machine

A.1. Use of groups for the Authorization Session State Machine

Section 8.1 in [RFC6733] defines a set of finite state machines, representing the life cycle of Diameter sessions, and which MUST be observed by all Diameter implementations that make use of the authentication and/or authorization portion of a Diameter application. This section defines, as example, additional state transitions related to the processing of the group commands which may impact multiple sessions.

The group membership is session state and therefore only those state machines from [RFC6733] in which the server is maintaining session state are relevant in this document. As in [RFC6733], the term Service-Specific below refers to a message defined in a Diameter application (e.g., Mobile IPv4, NASREQ).

The following state machine is observed by a client when state is maintained on the server. State transitions which are unmodified from [RFC6733] are not repeated here.

The Diameter group command in the following tables is differentiated from a single-session related command by a preceding 'G' (Group). A Group Re-Auth Request, which applies to one or multiple session groups, has been exemplarily described in Section 6.1. Such Group RAR command is denoted as 'GRAR' in the following table. The same notation applies to other commands as per [RFC6733].

CLIENT, STATEFUL				
State	Event		Action	New State
Idle	Client or Device Requests access		Send service specific auth req optionally including groups	Pending
Open	GASR received with Group-Response-Action = ALL_GROUPS, session is assigned to received group(s) and client will comply with request to end the session		Send GASA with Result-Code = SUCCESS, Send GSTR.	Discon
Open	GASR received with Group-Response-Action = PER_GROUPS, session is assigned to received group(s) and client will comply with request to end the session		Send GASA with Result-Code = SUCCESS, Send GSTR per group	Discon
Open	GASR received with Group-Response-Action = PER_SESSION, session is assigned to received group(s) and		Send GASA with Result-Code = SUCCESS, Send STR	Discon

	client will comply with request to end the session	per session	
Open	GASR received, client will not comply with request to end all session in received group(s)	Send GASA with Result-Code != SUCCESS	Open
Discon	GSTA Received	Discon. user/device	Idle
Open	GRAR received with Group-Response-Action = ALL_GROUPS, session is assigned to received group(s) and client will perform subsequent re-auth	Send GRAA, Send service specific group re-auth req	Pending
Open	GRAR received with Group-Response-Action = PER_GROUP, session is assigned to received group(s) and client will perform subsequent re-auth	Send GRAA, Send service specific group re-auth req per group	Pending
Open	GRAR received with Group-Response-Action = PER_SESSION, session is assigned to received group(s) and client will perform subsequent re-auth	Send GRAA, Send service specific re-auth req per session	Pending
Open	GRAR received and client will not perform subsequent re-auth	Send GRAA with Result-Code != SUCCESS, Discon. user/device	Idle
Pending	Successful service-specific group re-authorization answer received.	Provide service	Open
Pending	Failed service-specific group re-authorization answer	Discon. user/device	Idle

received.

The following state machine is observed by a server when it is maintaining state for the session. State transitions which are unmodified from [RFC6733] are not repeated here.

SERVER, STATEFUL			
State	Event	Action	New State
Idle	Service-specific authorization request received, and user is authorized	Send successful service specific answer optionally including groups	Open
Open	Server wants to terminate group(s)	Send GASR	Discon
Discon	GASA received	Cleanup	Idle
Any	GSTR received	Send GSTA, Cleanup	Idle
Open	Server wants to reauth group(s)	Send GRAR	Pending
Pending	GRAA received with Result-Code = SUCCESS	Update session(s)	Open
Pending	GRAA received with Result-Code != SUCCESS	Cleanup session(s)	Idle
Open	Service-specific group re-authorization request received and user is authorized	Send successful service specific group re-auth answer	Open
Open	Service-specific group re-authorization request received and user is not authorized	Send failed service specific group re-auth answer, cleanup	Idle

Authors' Addresses

Mark Jones

Email: mark@azu.ca

Marco Liebsch

Email: marco.liebsch@neclab.eu

Lionel Morand

Email: lionel.morand@orange.com

Diameter Maintenance and Extensions (DIME)
Internet-Draft
Intended status: Standards Track
Expires: March 19, 2018

M. Jones

M. Liebsch

L. Morand
September 15, 2017

Diameter Group Signaling
draft-ietf-dime-group-signaling-09.txt

Abstract

In large network deployments, a single Diameter node can support over a million concurrent Diameter sessions. Recent use cases have revealed the need for Diameter nodes to apply the same operation to a large group of Diameter sessions concurrently. The Diameter base protocol commands operate on a single session so these use cases could result in many thousands of command exchanges to enforce the same operation on each session in the group. In order to reduce signaling, it would be desirable to enable bulk operations on all (or part of) the sessions managed by a Diameter node using a single or a few command exchanges. This document specifies the Diameter protocol extensions to achieve this signaling optimization.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 19, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. Protocol Overview	4
3.1. Building and Modifying Session Groups	4
3.2. Issuing Group Commands	4
3.3. Permission Considerations	4
4. Protocol Description	6
4.1. Session Grouping Capability Discovery	7
4.1.1. Explicit Capability Discovery	7
4.1.2. Implicit Capability Discovery	7
4.2. Session Grouping	7
4.2.1. Group assignment at session initiation	8
4.2.2. Removing a session from a session group	11
4.2.3. Mid-session group assignment modifications	12
4.3. Deleting a Session Group	13
4.4. Performing Group Operations	13
4.4.1. Sending Group Commands	13
4.4.2. Receiving Group Commands	14
4.4.3. Error Handling for Group Commands	14
4.4.4. Single-Session Fallback	15
5. Operation with Proxy Agents	15
6. Commands Formatting	16
6.1. Formatting Example: Group Re-Auth-Request	16
7. Attribute-Value-Pairs (AVP)	17
7.1. Session-Group-Info AVP	17
7.2. Session-Group-Control-Vector AVP	18
7.3. Session-Group-Id AVP	18
7.4. Group-Response-Action AVP	19
7.5. Session-Group-Capability-Vector AVP	19
8. Result-Code AVP Values	19
9. IANA Considerations	19
9.1. AVP Codes	20
10. Security Considerations	20
11. Acknowledgments	21
12. Normative References	21
Appendix A. Session Management -- Exemplary Session State	

Machine 21
 A.1. Use of groups for the Authorization Session State Machine 21
 Authors' Addresses 26

1. Introduction

In large network deployments, a single Diameter node can support over a million concurrent Diameter sessions. Recent use cases have revealed the need for Diameter nodes to apply the same operation to a large group of Diameter sessions concurrently. For example, a policy decision point may need to modify the authorized quality of service for all active users having the same type of subscription. The Diameter base protocol commands operate on a single session so these use cases could result in many thousands of command exchanges to enforce the same operation on each session in the group. In order to reduce signaling, it would be desirable to enable bulk operations on all (or part of) the sessions managed by a Diameter node using a single or a few command exchanges.

This document describes mechanisms for grouping Diameter sessions and applying Diameter commands, such as performing re-authentication, re-authorization, termination and abortion of sessions to a group of sessions. This document does not define a new Diameter application. Instead it defines mechanisms, commands and AVPs that may be used by any Diameter application that requires management of groups of sessions.

These mechanisms take the following design goals and features into account:

- o Minimal impact to existing applications
- o Extension of existing commands' Command Code Format (CCF) with optional AVPs to enable grouping and group operations
- o Fallback to single session operation
- o Implicit discovery of capability to support grouping and group operations in case no external mechanism is available to discover a Diameter peer's capability to support session grouping and session group operations

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document uses terminology defined in [RFC6733].

3. Protocol Overview

3.1. Building and Modifying Session Groups

Client and Server can assign a new Diameter session to a group, e.g. in case the subscription profile of the associated user has similar characteristics as the profile of other users whose Diameter session has been assigned to one or multiple groups. A single command can be issued and applied to all sessions associated with such group(s), e.g. to adjust common profile or policy settings.

The assignment of a Diameter session to a group can be changed mid-session. For example, if a user's subscription profile changes mid-session, a Diameter server may remove the session from its current group and assign the session to a different group that is more appropriate for the new subscription profile.

In case of mobile users, the user's session may get transferred to a new Diameter client during handover and assigned to a different group, which is maintained at the new Diameter client, mid-session.

A session group, which has sessions assigned, can be deleted, e.g. due to a change in multiple users' subscription profile so that the group's assigned sessions do not share certain characteristics anymore. Deletion of such group requires subsequent individual treatment of each of the assigned sessions. A node may decide to assign some of these sessions to any other existing or new group.

3.2. Issuing Group Commands

Changes in the network condition may result in the Diameter server's decision to close all sessions in a given group. The server issues a single Session Termination Request (STR) command, identifying the group of sessions which are to be terminated. The Diameter client treats the STR as group command and initiates termination of all sessions associated with the identified group. Subsequently, the client confirms successful termination of these sessions to the server by sending a single Session Termination Answer (STA) command, which includes the identifier of the group.

3.3. Permission Considerations

Permission considerations in the context of this draft apply to the permission of Diameter nodes to build new session groups, to assign/remove a session to/from a session group and to delete an existing session group.

This specification follows the most flexible model where both, a Diameter client and a Diameter server can create a new group and assign a new identifier to that session group. When a Diameter node decides to create a new session group, e.g. to group all sessions which share certain characteristics, the node builds a session group identifier according to the rules described in Section 7.3 and becomes the owner of the group. This specification does not constrain the permission to add or remove a session to/from a session group to the group owner, instead each node can add a session to any known group or remove a session from a group. A session group is deleted and its identifier released after the last session has been removed from the session group. Also the modification of groups in terms of moving a session from one session group to a different session group is permitted to any Diameter node. A Diameter node can delete a session group and its group identifier mid-session, resulting in individual treatment of the sessions which have been previously assigned to the deleted group. Prerequisite for deletion of a session group is that the Diameter node created the session beforehand, hence the node became the group owner.

The enforcement of more constrained permissions is left to the specification of a particular group signaling enabled Diameter application and compliant implementations of such application MUST enforce the associated permission model. Details about enforcing a more constraint permission model are out of scope of this specification. For example, a more constrained model could require that a client MUST NOT remove a session from a group which is owned by the server.

The following table depicts the permission considerations as per the present specification:

Operation	Server	Client
Create a new Session Group (Diameter node becomes the group owner)	X	X
Assign a Session to an owned Session Group	X	X
Assign a Session to a non-owned Session Group	X	X
Remove a Session from an owned Session Group	X	X
Remove a Session from a non-owned Session Group	X	X
Remove a Session from a Session Group where the Diameter node created the assignment	X	X
Remove a Session from a Session Group where a different Diameter node created the assignment		
Overrule a different Diameter node's group assignment *)		
Delete a Session Group which is owned by the Diameter node	X	X
Delete a Session Group which is not owned by the Diameter node		

Default Permission as per this Specification

*) Editors' note: The protocol specification in this document does not consider overruling a node's assignment of a session to a session group. Here, overruling is to be understood as a node changing the group(s) assignment as per the node's request. Group signaling enabled applications may take such protocol support and associated protocol semantics into account in their specification. One exception is adopted in this specification, which allows a Diameter server to reject a group assignment as per the client's request.

4. Protocol Description

4.1. Session Grouping Capability Discovery

Diameter nodes SHOULD assign a session to a session group and perform session group operations with a node only after having ensured that the node announced associated support beforehand.

4.1.1. Explicit Capability Discovery

New Diameter applications may consider support for Diameter session grouping and for performing group commands during the standardization process. Such applications provide intrinsic discovery for the support of group commands and announce this capability through the assigned application ID.

System- and deployment-specific means, as well as out-of-band mechanisms for capability exchange can be used to announce nodes' support for session grouping and session group operations. In such case, the optional Session-Group-Capability-Vector AVP, as described in Section 4.1.2 can be omitted in Diameter messages being exchanged between nodes.

4.1.2. Implicit Capability Discovery

If no explicit mechanism for capability discovery is deployed to enable Diameter nodes to learn about nodes' capability to support session grouping and group commands, a Diameter node SHOULD append the Session-Group-Capability-Vector AVP to any Diameter messages exchanged with its nodes to announce its capability to support session grouping and session group operations. Implementations following the specification as per this document set the `BASE_SESSION_GROUP_CAPABILITY` flag of the Session-Group-Capability-Vector AVP.

When a Diameter node receives at least one Session-Group-Capability-Vector AVP from a node with the `BASE_SESSION_GROUP_CAPABILITY` flag set, the Diameter node maintains a log to remember the node's capability to support group commands.

4.2. Session Grouping

This specification does not limit the number of session groups, to which a single session is assigned. It is left to the application to determine the policy of session grouping. In case an application facilitates a session to belong to multiple session groups, the application MUST maintain consistency of associated application session states for these multiple session groups.

Either Diameter node (client or server) can initiate the assignment of a session to a single or multiple session groups. Modification of a group by removing or adding a single or multiple user sessions can be initiated and performed mid-session by either Diameter node. Diameter AAA applications typically assign client and server roles to the Diameter nodes, which are referred to as relevant Diameter nodes to utilize session grouping and issue group commands. Section 5 describes particularities about session grouping and performing group commands when relay agents or proxies are deployed.

Diameter nodes, which are group-aware, MUST store and maintain an entry about the group assignment together with a session's state. A list of all known session groups should be locally maintained on each node, each group pointing to individual sessions being assigned to the group. A Diameter node MUST also keep a record about sessions, which have been assigned to a session group by itself.

4.2.1. Group assignment at session initiation

To assign a session to a group at session initiation, a Diameter client sends a service specific request, e.g. NASREQ AA-Request [RFC4005], containing one or more session group identifiers. Each of these groups MUST be identified by a unique Session-Group-Id contained in a separate Session-Group-Info AVP as specified in Section 7.

The client may choose one or multiple session groups from a list of existing session groups. Alternatively, the client may decide to create a new group to which the session is assigned and identify itself in the <DiameterIdentity> portion of the Session-Group-Id AVP as per Section 7.3

The client MUST set the SESSION_GROUP_ALLOCATION_ACTION flag of the Session-Group-Control-Vector AVP in each appended Session-Group-Info AVP to indicate that the session contained in the request should be assigned to the identified session group.

The client may also indicate in the request that the server is responsible for the assignment of the session in one or multiple sessions owned by the server. In such a case, the client MUST include the Session-Group-Info AVP in the request including the Session-Group-Control-Vector AVP with the SESSION_GROUP_ALLOCATION_ACTION flag set but no Session-Group-Id AVP.

If the Diameter server receives a command request from a Diameter client and the command comprises at least one Session-Group-Info AVP having the SESSION_GROUP_ALLOCATION_ACTION flag in the Session-Group-Control-Vector AVP set, the server can accept or reject the request

for group assignment. Reasons for rejection may be e.g. lack of resources for managing additional groups. When rejected, the session MUST NOT be assigned to any session group.

If the Diameter server accepts the client's request for a group assignment, the server MUST assign the new session to each of the one or multiple identified session groups when present in the Session-Group-Info AVP. In case one or multiple identified session groups are not already stored by the server, the server MUST store the new identified group(s) to its local list of known session groups. When sending the response to the client, e.g. a service-specific auth response as per NASREQ AA-Answer [RFC4005], the server MUST include all Session-Group-Info AVPs as received in the client's request.

In addition to the one or multiple session groups identified in the client's request, the server may decide to assign the new session to one or multiple additional groups. In such a case, the server MUST add to the response the additional Session-Group-Info AVPs, each identifying a session group to which the new session is assigned by the server. Each of the Session-Group-Info AVP added by the server MUST have the SESSION_GROUP_ALLOCATION_ACTION flag set in the Session-Group-Control-Vector AVP set.

If the Diameter server rejects the client's request for a group assignment, the server sends the response to the client, e.g. a service-specific auth response as per NASREQ AA-Answer [RFC4005], and MUST include all Session-Group-Info AVPs as received in the client's request (if any) while clearing the SESSION_GROUP_ALLOCATION_ACTION flag of the Session-Group-Control-Vector AVP. The server MAY accept the client's request for the identified session but refuse the session's assignment to any session group. The server sends the response to the client indicating success in the result code. In such case the session is treated as single session without assignment to any session group by the Diameter nodes.

If the Diameter server accepts the client's request for a group assignment, but the assignment of the session to one or some of the multiple identified session groups fails, the session group assignment is treated as failure. In such case the session is treated as single session without assignment to any session group by the Diameter nodes. The server sends the response to the client and MAY include as information to the client only those Session-Group-Info AVPs for which the group assignment failed. The SESSION_GROUP_ALLOCATION_ACTION flag of included Session-Group-Info AVPs MUST be cleared.

If the Diameter server receives a command request from a Diameter client and the command comprises one or multiple Session-Group-Info

AVPs and none of them includes a Session-Group-Id AVP, the server MAY decide to assign the session to one or multiple session groups. For each session group, to which the server assigns the new session, the server includes a Session-Group-Info AVP with the Session-Group-Id AVP identifying a session group in the response sent to the client. Each of the Session-Group-Info AVPs included by the server MUST have the SESSION_GROUP_ALLOCATION_ACTION flag of the Session-Group-Control-Vector AVP set.

If the Diameter server receives a command request from a Diameter client and the command does not contain any Session-Group-Info AVP, the server MUST NOT assign the new session to any session group but treat the request as for a single session. The server MUST NOT return any Session-Group-Info AVP in the command response.

If the Diameter client receives a response to its previously issued request from the server and the response comprises at least one Session-Group-Info AVP having the SESSION_GROUP_ALLOCATION_ACTION flag of the associated Session-Group-Control-Vector AVP set, the client MUST add the new session to all session groups as identified in the one or multiple Session-Group-Info AVPs. If the Diameter client fails to add the session to one or more session groups as identified in the one or multiple Session-Group-Info AVPs, the client MUST terminate the session. The client MAY send a subsequent request for session initiation to the server without requesting the assignment of the session to a session group

If the Diameter client receives a response to its previously issued request from the server and the one or more Session-Group-Info AVPs have the SESSION_GROUP_ALLOCATION_ACTION flag of the associated Session-Group-Control-Vector AVP cleared, the client MUST terminate the assignment of the session to the one or multiple groups. If the response from the server indicates success in the result code but solely the assignment of the session to a session group has been rejected by the server, the client treats the session as single session without group assignment.

A Diameter client, which sent a request for session initiation to a Diameter server and appended a single or multiple Session-Group-Id AVPs but cannot find any Session-Group-Info AVP in the associated response from the Diameter server proceeds as if the request was processed for a single session. When the Diameter client is confident that the Diameter server supports session grouping and group signaling, the Diameter client SHOULD NOT retry to request group assignment for this session, but MAY try to request group assignment for other new sessions.

4.2.2. Removing a session from a session group

When a Diameter client decides to remove a session from a particular session group, the client sends a service-specific re-authorization request to the server and adds one Session-Group-Info AVP to the request for each session group, from which the client wants to remove the session. The session, which is to be removed from a group, is identified in the Session-Id AVP of the command request. The `SESSION_GROUP_ALLOCATION_ACTION` flag of the Session-Group-Control-Vector AVP in each Session-Group-Info AVP **MUST** be cleared to indicate removal of the session from the session group identified in the associated Session-Group-id AVP.

When a Diameter client decides to remove a session from all session groups, to which the session has been previously assigned, the client sends a service-specific re-authorization request to the server and adds a single Session-Group-Info AVP to the request which has the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared and the Session-Group-Id AVP omitted. The session, which is to be removed from all groups, to which the session has been previously assigned, is identified in the Session-Id AVP of the command request.

If the Diameter server receives a request from the client which has at least one Session-Group-Info AVP appended with the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared, the server **MUST** remove the session from the session group identified in the associated Session-Group-Id AVP. If the request comprises at least one Session-Group-info AVP with the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared and no Session-Id AVP present, the server **MUST** remove the session from all session groups to which the session has been previously assigned. The server **MUST** include in its response to the requesting client all Session-Group-Id AVPs as received in the request.

When the Diameter server decides to remove a session from one or multiple particular session groups or from all session groups to which the session has been assigned beforehand, the server sends a Re-Authorization Request (RAR) or a service-specific server-initiated request to the client, indicating the session in the Session-Id AVP of the request. The client sends a Re-Authorization Answer (RAA) or a service-specific answer to respond to the server's request. The client subsequently sends service-specific re-authorization request containing one or multiple Session-Group-Info AVPs, each indicating a session group, to which the session had been previously assigned. To indicate removal of the indicated session from one or multiple session groups, the server sends a service-specific auth response to the client, containing a list of Session-Group-Info AVPs with the `SESSION_GROUP_ALLOCATION_ACTION` flag cleared and the Session-Group-Id AVP identifying the session group, from which the session should be

removed. The server MAY include to the service-specific auth response a list of Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP identifying session groups to which the session remains subscribed. In case the server decides to remove the identified session from all session groups, to which the session has been previously assigned, the server includes in the service-specific auth response at least one Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and Session-Group-Id AVP absent.

4.2.3. Mid-session group assignment modifications

Either Diameter node (client or server) can modify the group membership of an active Diameter session according to the specified permission considerations.

To update an assigned group mid-session, a Diameter client sends a service-specific re-authorization request to the server, containing one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP present, identifying the session group to which the session should be assigned. With the same message, the client may send one or multiple Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and the Session-Group-Id AVP identifying the session group from which the identified session is to be removed. To remove the session from all previously assigned session groups, the client includes at least one Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and no Session-Group-Id AVP present. When the server received the service-specific re-authorization request, it MUST update its locally maintained view of the session groups for the identified session according to the appended Session-Group-Info AVPs. The server sends a service-specific auth response to the client containing one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP identifying the new session group to which the identified session has been assigned.

When a Diameter server enforces an update to the assigned groups mid-session, it sends a Re-Authorization Request (RAR) message to the client identifying the session, for which the session group lists are to be updated. The client responds with a Re-Authorization Answer (RAA) message. The client subsequently sends a service-specific re-authorization request containing one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag set and the Session-Group-Id AVP identifying the session group to which the session had been previously assigned. The server responds with a service-specific auth response and includes one or multiple Session-Group-Info AVP with the SESSION_GROUP_ALLOCATION_ACTION flag set and

the Session-Group-Id AVP identifying the session group, to which the identified session is to be assigned. With the same response message, the server may send one or multiple Session-Group-Info AVPs with the SESSION_GROUP_ALLOCATION_ACTION flag cleared and the Session-Group-Id AVP identifying the session groups from which the identified session is to be removed. When server wants to remove the session from all previously assigned session groups, it sends at least one Session-Group-Info AVP with the response having the SESSION_GROUP_ALLOCATION_ACTION flag cleared and no Session-Group-Id AVP present.

4.3. Deleting a Session Group

To delete a session group and release the associated Session-Group-Id value, the owner of a session group appends a single Session-Group-Info AVP having the SESSION_GROUP_STATUS_IND flag cleared and the Session-Group-Id AVP identifying the session group, which is to be deleted. The SESSION_GROUP_ALLOCATION_ACTION flag of the associated Session-Group-Control-Vector AVP MUST be cleared.

4.4. Performing Group Operations

4.4.1. Sending Group Commands

Either Diameter node (client or server) can request the recipient of a request to process an associated command for all sessions being assigned to one or multiple groups by identifying these groups in the request. The sender of the request appends for each group, to which the command applies, a Session-Group-Info AVP including the Session-Group-Id AVP to identify the associated session group. Both, the SESSION_GROUP_ALLOCATION_ACTION flag as well as the SESSION_GROUP_STATUS_IND flag MUST be set.

If the CCF of the request mandates a Session-Id AVP, the Session-Id AVP MUST identify one of the single sessions which is assigned to at least one of the groups being identified in the appended Session-Group-Id AVPs.

The sender of the request MUST indicate to the receiver how follow up message exchanges should be performed by appending a single instance of the Group-Response-Action AVP. Even if the request includes multiple instances of a Session-Group-Info AVP, the request MUST NOT comprise more than a single instance of a Group-Response-Action AVP. If the sender wants the receiver to perform follow up exchanges with a single command for all impacted groups, the sender sets the value of the Group-Response-Action AVP to ALL_GROUPS (1). If follow up message exchanges should be performed on a per-group basis in case multiple groups are identified in the group command, the value of the

Group-Response-Action AVP is set to PER_GROUP (2). A value set to PER_SESSION (3) indicates to the receiver that all follow up exchanges should be performed using a single message for each impacted session.

If the sender sends a request including the Group-Response-Action AVP set to ALL_GROUPS (1) or PER_GROUP (2), it MUST expect some delay before receiving the corresponding answer(s) as the answer(s) will only be sent back when the request is processed for all the sessions or all the session of a session group. If the process of the request is delay-sensitive, the sender SHOULD NOT set the Group-Response-Action AVP to ALL_GROUPS (1) or PER_GROUP (2). If the answer can be sent before the complete process of the request for all the sessions or if the request timeout timer is high enough, the sender MAY set the Group-Response-Action AVP to ALL_GROUPS (1) or PER_GROUP (2).

If the sender wants the receiver of the request to process the associated command solely for a single session, the sender does not append any group identifier, but identifies the relevant session in the Session-Id AVP.

4.4.2. Receiving Group Commands

A Diameter node receiving a request to process a command for a group of sessions, identifies the relevant groups according to the appended Session-Group-Id AVP in the Session-Group-Info AVP and processes the group command according to the appended Group-Response-Action AVP . If the received request identifies multiple groups in multiple appended Session-Group-Id AVPs, the receiver SHOULD process the associated command for each of these groups. If a session has been assigned to more than one of the identified groups, the receiver MUST process the associated command only once per session.

4.4.3. Error Handling for Group Commands

When a Diameter node receives a request to process a command for one or more session groups and the result of processing the command is an error that applies to all sessions in the identified groups, an associated protocol error MUST be returned to the source of the request. In such case, the sender of the request MUST fall back to single-session processing and the session groups, which have been identified in the group command, MUST be deleted according to the procedure described in Section 4.3.

When a Diameter node receives a request to process a command for one or more session groups and the result of processing the command succeeds for some sessions identified in one or multiple session groups, but fails for one or more sessions, the Result-Code AVP in

the response message SHOULD indicate `DIAMETER_LIMITED_SUCCESS` as per Section 7.1.2 of [RFC6733].

In case of limited success, the sessions, for which the processing of the group command failed, MUST be identified using a Failed-AVP AVP as per Section 7.5 of [RFC6733]. The sender of the request MUST fall back to single-session operation for each of the identified sessions, for which the group command failed. In addition, each of these sessions MUST be removed from all session groups to which the group command applied. To remove sessions from a session group, the Diameter client performs the procedure described in Section 4.2.2.

4.4.4. Single-Session Fallback

Either Diameter node can fall back to single session operation by ignoring and omitting the optional group session-specific AVPs. Fallback to single-session operation is performed by processing the Diameter command solely for the session identified in the mandatory Session-Id AVP. In such case, the response to the group command MUST NOT identify any group but identify solely the single session for which the command has been processed.

5. Operation with Proxy Agents

In case of a present stateful Proxy Agent between a Diameter client and a Diameter server, this specification assumes that the Proxy Agent is aware of session groups and session group handling. The Proxy MUST update and maintain consistency of its local session states as per the result of the group commands which are operated between a Diameter client and a server. In such case, the Proxy Agent MUST act as a Diameter server in front of the Diameter client and MUST act as a Diameter client in front of the Diameter server. Therefore, the client and server behavior described in Section 4 applies respectively to the stateful Proxy Agent.

In case a stateful Proxy Agent manipulates session groups, it MUST maintain consistency of session groups between a client and a server. This applies to a deployment where the Proxy Agent utilizes session grouping and performs group operations with, for example, a Diameter server, whereas the Diameter client is not aware of session groups. In such case the Proxy Agent must reflect the states associated with the session groups as individual session operations towards the client and ensure the client has a consistent view of each session. The same applies to a deployment where all nodes, the Diameter client and server, as well as the Proxy Agent are group-aware but the Proxy Agent manipulates groups, e.g. to adopt different administrative policies that apply to the client's domain and the server's domain.

Stateless Proxy Agents do not maintain any session state (only transaction state are maintained). Consequently, the notion of session group is transparent for any stateless Proxy Agent present between a Diameter client and a Diameter server handling session groups. Session group related AVPs being defined as optional AVP SHOULD be ignored by stateless Proxy Agents and SHOULD NOT be removed from the Diameter commands. If they are removed by the Proxy Agent for any reason, the Diameter client and Diameter server will discover the absence the related session group AVPs and will fall back to single-session processing, as described in Section 4.

6. Commands Formatting

This document does not specify new Diameter commands to enable group operations, but relies on command extensibility capability provided by the Diameter Base protocol. This section provides the guidelines to extend the CCF of existing Diameter commands with optional AVPs to enable the recipient of the command applying the command to all sessions associated with the identified group(s).

6.1. Formatting Example: Group Re-Auth-Request

A request for re-authentication of one or more groups of users is issued by appending one or multiple Session-Group-Id AVP(s), as well as a single instance of a Group-Response-Action AVP to the Re-Auth-Request (RAR). The one or multiple Session-Group-Id AVP(s) identify the associated group(s) for which the group re-authentication has been requested. The Group-Response-Action AVP identifies the expected means to perform and respond to the group command. The recipient of the group command initiates re-authentication for all users associated with the identified group(s). Furthermore, the sender of the group re-authentication request appends a Group-Response-Action AVP to provide more information to the receiver of the command about how to accomplish the group operation.

The value of the mandatory Session-Id AVP MUST identify a session associated with a single user, which is assigned to at least one of the groups being identified in the appended Session-Group-Id AVPs.


```

<RAR> ::= < Diameter Header: 258, REQ, PXY >
        < Session-Id >
        { Origin-Host }
        { Origin-Realm }
        { Destination-Realm }
        { Destination-Host }
        { Auth-Application-Id }
        { Re-Auth-Request-Type }
        [ User-Name ]
        [ Origin-State-Id ]
        * [ Proxy-Info ]
        * [ Route-Record ]
        [ Session-Group-Capability-Vector ]
        * [ Session-Group-Info ]
        [ Group-Response-Action ]
        * [ AVP ]
    
```

7. Attribute-Value-Pairs (AVP)

Attribute Name	AVP Code	Value Type	AVP Flag rules			
			MUST	MAY	SHOULD NOT	MUST NOT
Session-Group-Info	TBD1	Grouped		P		V
Session-Group-Control-Vector	TBD2	Unsigned32		P		V
Session-Group-Id	TBD3	OctetString		P		V
Group-Response-Action	TBD4	Unsigned32		P		V
Session-Group-Capability-Vector	TBD5	Unsigned32		P		V

AVPs for the Diameter Group Signaling

7.1. Session-Group-Info AVP

The Session-Group-Info AVP (AVP Code TBD1) is of type Grouped. It contains the identifier of the session group as well as an indication of the node responsible for session group identifier assignment.

```

Session-Group-Info ::= < AVP Header: TBD1 >
                        < Session-Group-Control-Vector >
                        [ Session-Group-Id ]
                        * [ AVP ]
    
```

7.2. Session-Group-Control-Vector AVP

The Session-Group-Control-Vector AVP (AVP Code TBD2) is of type Unsigned32 and contains a 32-bit flags field to control the group assignment at session-group aware nodes.

The following capabilities are defined in this document:

`SESSION_GROUP_ALLOCATION_ACTION (0x00000001)`

This flag indicates the action to be performed for the identified session. When this flag is set, it indicates that the identified Diameter session is to be assigned to the session group as identified by the Session-Group-Id AVP or the session's assignment to the session group identified in the Session-Group-Id AVP is still valid. When the flag is cleared, the identified Diameter session is to be removed from at least one session group. When the flag is cleared and the Session-Group-Info AVP identifies a particular session group in the associated Session-Group-Id AVP, the session is to be removed solely from the identified session group. When the flag is cleared and the Session-Group-Info AVP does not identify a particular session group (Session-Group-Id AVP is absent), the identified Diameter session is to be removed from all session groups, to which it has been previously assigned.

`SESSION_GROUP_STATUS_IND (0x00000010)`

This flag indicates the status of the session group identified in the associated Session-Group-Id AVP. The flag is set when the identified session group has just been created or is still active. If the flag is cleared, the identified session group is deleted and the associated Session-Group-Id is released. If the Session-Group-Info AVP does not comprise a Session-Group-Id AVP, this flag is meaningless and MUST be ignored by the receiver.

7.3. Session-Group-Id AVP

The Session-Group-Id AVP (AVP Code TBD3) is of type UTF8String and identifies a group of Diameter sessions.

The Session-Group-Id MUST be globally and eternally unique, as it is meant to uniquely identify a group of Diameter sessions without reference to any other information.

The default format of the Session-Group-id MUST comply to the format recommended for a Session-Id, as defined in the section 8.8 of the [RFC6733]. The <DiameterIdentity> portion of the Session-Group-Id MUST identify the Diameter node, which owns the session group.

7.4. Group-Response-Action AVP

The Group-Response-Action AVP (AVP Code TBD4) is of type Unsigned32 and contains a 32-bit address space representing values indicating how the node SHOULD issue follow up exchanges in response to a command which impacts multiple sessions. The following values are defined by this application:

ALL_GROUPS (1)

Follow up exchanges should be performed with a single message exchange for all impacted groups.

PER_GROUP (2)

Follow up exchanges should be performed with a message exchange for each impacted group.

PER_SESSION (3)

Follow up exchanges should be performed with a message exchange for each impacted session.

7.5. Session-Group-Capability-Vector AVP

The Session-Group-Capability-Vector AVP (AVP Code TBD5) is of type Unsigned32 and contains a 32-bit flags field to indicate capabilities in the context of session-group assignment and group operations.

The following capabilities are defined in this document:

BASE_SESSION_GROUP_CAPABILITY (0x00000001)

This flag indicates the capability to support session grouping and session group operations according to this specification.

8. Result-Code AVP Values

This document does not define new Result-Code [RFC6733] values for existing applications, which are extended to support group commands. Specification documents of new applications, which will have intrinsic support for group commands, may specify new Result-Codes.

9. IANA Considerations

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

9.1. AVP Codes

This specification requires IANA to register the following new AVPs from the AVP Code namespace defined in [RFC6733].

- o Session-Group-Info
- o Session-Group-Control-Vector
- o Session-Group-Id
- o Group-Response-Action
- o Session-Group-Capability-Vector

The AVPs are defined in Section 7.

10. Security Considerations

The security considerations of the Diameter protocol itself are discussed in [RFC6733]. Use of the AVPs defined in this document MUST take into consideration the security issues and requirements of the Diameter base protocol. In particular, the Session-Group-Info AVP (including the Session-group-Control-Vector and the Session-Group-Id AVPs) should be considered as a security-sensitive AVPs in the same manner than the Session-Id AVP in the Diameter base protocol [RFC6733].

The management of session groups relies upon the existing trust relationship between the Diameter client and the Diameter server managing the groups of sessions. This document defines a mechanism that allows a client or a server to act on multiple sessions at the same time using only one command. if the Diameter client or server is compromised, an attacker could launch DoS attacks by terminating a large number of sessions with a limited set of commands using the session group management concept.

According to the Diameter base protocol [RFC6733], transport connections between Diameter peers are protected by TLS/TCP, DTLS/SCTP or alternative security mechanisms that are independent of Diameter, such as IPsec. However, the lack of end-to-end security features makes it difficult to establish trust in the session group related information received from non-adjacent nodes. Any Diameter agent in the message path can potentially modify the content of the message and therefore the information sent by the Diameter client or the server. The DIME working group is currently working on solutions for providing end-to-end security features. When available, these features should enable the establishment of trust relationship

between non-adjacent nodes and the security required for session group management would normally rely on this end-to-end security. However, there is no assumption in this document that such end-to-end security mechanism will be available. It is only assume that the solution defined on this document relies on the security framework provided by the Diameter based protocol.

In some cases, a Diameter Proxy agent can act on behalf of a client or server. In such a case, the security requirements that normally apply to a client (or a server) apply equally to the Proxy agent.

11. Acknowledgments

The authors of this document want to thank Ben Campbell and Eric McMurry for their valuable comments to early versions of this draft. Furthermore, authors thank Steve Donovan and Mark Bales for the thorough review and comments on advanced versions of the WG document, which helped a lot to improve this specification.

12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4005] Calhoun, P., Zorn, G., Spence, D., and D. Mitton, "Diameter Network Access Server Application", RFC 4005, DOI 10.17487/RFC4005, August 2005, <<https://www.rfc-editor.org/info/rfc4005>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<https://www.rfc-editor.org/info/rfc6733>>.

Appendix A. Session Management -- Exemplary Session State Machine

A.1. Use of groups for the Authorization Session State Machine

Section 8.1 in [RFC6733] defines a set of finite state machines, representing the life cycle of Diameter sessions, and which MUST be observed by all Diameter implementations that make use of the authentication and/or authorization portion of a Diameter application. This section defines, as example, additional state transitions related to the processing of the group commands which may impact multiple sessions.

The group membership is session state and therefore only those state machines from [RFC6733] in which the server is maintaining session state are relevant in this document. As in [RFC6733], the term Service-Specific below refers to a message defined in a Diameter application (e.g., Mobile IPv4, NASREQ).

The following state machine is observed by a client when state is maintained on the server. State transitions which are unmodified from [RFC6733] are not repeated here.

The Diameter group command in the following tables is differentiated from a single-session related command by a preceding 'G' (Group). A Group Re-Auth Request, which applies to one or multiple session groups, has been exemplarily described in Section 6.1. Such Group RAR command is denoted as 'GRAR' in the following table. The same notation applies to other commands as per [RFC6733].

CLIENT, STATEFUL				
State	Event		Action	New State
Idle	Client or Device Requests access		Send service specific auth req optionally including groups	Pending
Open	GASR received with Group-Response-Action = ALL_GROUPS, session is assigned to received group(s) and client will comply with request to end the session		Send GASA with Result-Code = SUCCESS, Send GSTR.	Discon
Open	GASR received with Group-Response-Action = PER_GROUPS, session is assigned to received group(s) and client will comply with request to end the session		Send GASA with Result-Code = SUCCESS, Send GSTR per group	Discon
Open	GASR received with Group-Response-Action = PER_SESSION, session is assigned to received group(s) and		Send GASA with Result-Code = SUCCESS, Send STR	Discon

	client will comply with request to end the session	per session	
Open	GASR received, client will not comply with request to end all session in received group(s)	Send GASA with Result-Code != SUCCESS	Open
Discon	GSTA Received	Discon. user/device	Idle
Open	GRAR received with Group-Response-Action = ALL_GROUPS, session is assigned to received group(s) and client will perform subsequent re-auth	Send GRAA, Send service specific group re-auth req	Pending
Open	GRAR received with Group-Response-Action = PER_GROUP, session is assigned to received group(s) and client will perform subsequent re-auth	Send GRAA, Send service specific group re-auth req per group	Pending
Open	GRAR received with Group-Response-Action = PER_SESSION, session is assigned to received group(s) and client will perform subsequent re-auth	Send GRAA, Send service specific re-auth req per session	Pending
Open	GRAR received and client will not perform subsequent re-auth	Send GRAA with Result-Code != SUCCESS, Discon. user/device	Idle
Pending	Successful service-specific group re-authorization answer received.	Provide service	Open
Pending	Failed service-specific group re-authorization answer	Discon. user/device	Idle

received.

The following state machine is observed by a server when it is maintaining state for the session. State transitions which are unmodified from [RFC6733] are not repeated here.

SERVER, STATEFUL			
State	Event	Action	New State
Idle	Service-specific authorization request received, and user is authorized	Send successful service specific answer optionally including groups	Open
Open	Server wants to terminate group(s)	Send GASR	Discon
Discon	GASA received	Cleanup	Idle
Any	GSTR received	Send GSTA, Cleanup	Idle
Open	Server wants to reauth group(s)	Send GRAR	Pending
Pending	GRAA received with Result-Code = SUCCESS	Update session(s)	Open
Pending	GRAA received with Result-Code != SUCCESS	Cleanup session(s)	Idle
Open	Service-specific group re-authorization request received and user is authorized	Send successful service specific group re-auth answer	Open
Open	Service-specific group re-authorization request received and user is not authorized	Send failed service specific group re-auth answer, cleanup	Idle

Authors' Addresses

Mark Jones

Email: mark@azu.ca

Marco Liebsch

Email: marco.liebsch@neclab.eu

Lionel Morand

Email: lionel.morand@orange.com

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 8, 2017

B. Campbell
S. Donovan, Ed.
Oracle
JJ. Trottin
Nokia
March 7, 2017

Diameter Load Information Conveyance
draft-ietf-dime-load-08

Abstract

RFC7068 describes requirements for Overload Control in Diameter. This includes a requirement to allow Diameter nodes to send "load" information, even when the node is not overloaded. RFC7683 (Diameter Overload Information Conveyance (DOIC)) solution describes a mechanism meeting most of the requirements, but does not currently include the ability to send load information. This document defines a mechanism for conveying of Diameter load information.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 8, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Terminology and Abbreviations	3
3. Conventions Used in This Document	4
4. Background	4
4.1. Differences between Load and Overload information	4
4.2. How is Load Information Used?	5
5. Solution Overview	6
5.1. Theory of Operation	8
6. Load Mechanism Procedures	10
6.1. Reporting Node Behavior	10
6.1.1. Endpoint Reporting Node Behavior	10
6.1.2. Agent Reporting Node Behavior	11
6.2. Reacting Node Behavior	12
6.3. Extensibility	13
6.4. Addition and Removal of Nodes	13
7. Attribute Value Pairs	14
7.1. Load AVP	14
7.2. Load-Type AVP	14
7.3. Load-Value AVP	14
7.4. SourceID AVP	15
7.5. Attribute Value Pair flag rules	15
8. Security Considerations	15
9. IANA Considerations	16
9.1. AVP Codes	16
9.2. New Registries	16
10. References	16
10.1. Normative References	16
10.2. Informative References	17
Appendix A. Topology Scenarios	17
A.1. No Agent	17
A.2. Single Agent	17
A.3. Multiple Agents	18
A.4. Linked Agents	19
A.5. Shared Server Pools	20
A.6. Agent Chains	20
A.7. Fully Meshed Layers	21
A.8. Partitions	21
A.9. Active-Standby Nodes	21
Authors' Addresses	22

1. Introduction

[RFC7068] describes requirements for Overload Control in Diameter [RFC6733]. The DIME working group has finished the Diameter Overload Information Conveyance (DOIC) mechanism [RFC7683]. As currently specified, DOIC fulfills some, but not all, of the requirements.

In particular, DOIC does not fulfill Req 23 and Req 24:

REQ 23: The solution MUST provide sufficient information to enable a load-balancing node to divert messages that are rejected or otherwise throttled by an overloaded upstream node to other upstream nodes that are the most likely to have sufficient capacity to process them.

REQ 24: The solution MUST provide a mechanism for indicating load levels, even when not in an overload condition, to assist nodes in making decisions to prevent overload conditions from occurring.

There are several other requirements in [RFC7068] that mention both overload and load information that are only partially fulfilled by DOIC.

The DIME working group explicitly chose not to fulfill these requirements in DOIC due to several reasons. A principal reason was that the working group did not agree on a general approach for conveying load information. It chose to progress the rest of DOIC, and deferred load information conveyance to a DOIC extension or a separate mechanism.

This document defines a mechanism that addresses the load-related requirements from RFC 7068.

2. Terminology and Abbreviations

AVP

Attribute Value Pair

DOIC

Diameter Overload Information Conveyance ([RFC7683])

Load

The relative usage of the Diameter message processing capacity of a Diameter node. A low load level indicates that the Diameter

node is under utilized. A high load level indicates that the node is closer to being fully utilized.

Offered Load

The actual traffic sent to the reporting node after overload abatement and routing decisions are made.

Reporting Node

Reporting Node: A Diameter node that generates a load report.

Reacting Node

Reacting Node: A Diameter node that acts upon a load report.

Routing Information

Routing Information referred to in this document can include the Routing and Peer tables defined in RFC 6733. It can also include other implementation specific tables used to store load information. This document does not define the structure of such tables.

3. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

RFC 2119 [RFC2119] interpretation does not apply for the above listed words when they are not used in all-caps format.

4. Background

4.1. Differences between Load and Overload information

Previous discussions of how to solve the load-related requirements in [RFC7068] have shown that people did not have an agreed-upon concept of how "load" information differs from "overload" information. While the two concepts are highly interrelated, there are two primary differences. First, a Diameter node always has a load. At any given time that load may be effectively zero, effectively fully loaded, or somewhere in between. In contrast, overload is an exceptional condition. A node only has overload information when it is in an overloaded state. Furthermore, the relationship between a node's load level and overload state at any given time may be vague. For example, a node may normally operate at a "fully loaded" level, but

still not be considered overloaded. Another node may declare itself to be "overloaded" even though it might not be fully "loaded".

Second, Overload information, in the form of a DOIC Overload Report (OLR) [RFC7683] indicates an explicit request for action on the part of the reacting node. That is, the OLR requests that the reacting node reduces the offered load -- the actual traffic sent to the reporting node after overload abatement and routing decisions are made -- by an indicated amount (by default), or as prescribed by the selected abatement algorithm. Effectively, DOIC provides a contract between the reporting node and the reacting node.

In contrast, load is informational. That is, load information can be considered a hint to the recipient node. That node may use the load information for load balancing purposes, as an input to certain overload abatement techniques, to make inferences about the likelihood that the sending node becomes overloaded in the immediate future, or for other purposes.

None of this prevents a Diameter node from deciding to reduce the offered load based on load information. The fundamental difference is that an overload report requires the reduction of offered load. It is also reasonable for a Diameter node to decide to increase the offered load based on load information.

4.2. How is Load Information Used?

[RFC7068] contemplates two primary uses for load information. Req 23 discusses how load information might be used when performing diversion as an overload abatement technique, as described in [RFC7683]. When a reacting node diverts traffic away from an overloaded node, it needs load information for the other candidates for that traffic in order to effectively load balance the diverted load between potential candidates. Otherwise, diversion has a greater potential to drive other nodes into overload.

Req 24 discusses how Diameter load information might be used when no overload condition currently exists. Diameter nodes can use the load information to make decisions to try to avoid overload conditions in the first place. Normal load-balancing falls into this category, but the diameter node can take other proactive steps as well.

If the loaded nodes are Diameter servers (or clients in the case of server-to-client transactions), both of these uses of load information should be accomplished by a Diameter node that performs server selection (selection of the Diameter endpoint to which the request is to be routed for processing). Typically, server selection is performed by a node (a client or an agent) that is an immediate

peer of the server. However, there are scenarios (see Appendix A) where a client or proxy that is not the immediate peer to the selected servers performs server selection. In this case, the client or proxy enforces the server selection by inserting a Destination-Host AVP.

For example, a Diameter node (e.g. client) can use a redirect agent to get candidate destination host addresses. The redirect agent might return several destination host addresses, from which the Diameter node selects one. The Diameter node can use load information received from these hosts to make the selection.

Just as load information can be used as part of server selection, it can also be used as input to the selection of the next-hop peer to which a request is to be routed.

It should be noted that a Diameter node will need to process both Load reports and Overload reports from the same Diameter node. The reacting node for the Overload report always has the responsibility to reduce the amount of Diameter traffic sent to the overloaded node. If, or how, the reacting node uses Load information to achieve this is left as an implementation decision.

5. Solution Overview

The mechanism defined here for the conveyance of load information is similar in some ways to the mechanism defined for DOIC and is different in other ways.

As with DOIC, load information is conveyed by piggy-backing the load AVPs on existing Diameter applications.

There are two primary differences. First, there is no capability negotiation process for load. The sender of the load information is sending it with the expectation that any supporting nodes will use it when making routing decisions. If there are no nodes that support the Load mechanism then the load information is ignored.

The second big difference between DOIC and Load is visibility of the DOIC or Load information within a Diameter network. DOIC information is sent end-to-end resulting in the ability of all nodes in the path of the answer message that carries the OC-OLR AVP to act on the information, although only one node actually consumes and reacts to the report. The DOIC overload reports remain in the message all the way from the reporting node to the node that is the target for the answer message.

For the Load mechanism there are two types of Load reports and only the first one is transmitted end-to-end.

The first type of load report is a HOST report which contains the load of the endpoint sending the answer message. This Load report is carried end-to-end to enable any nodes that make server selection decisions to use the load status of the sending endpoint as part of the server selection decision. Unlike with DOIC, more than one node may make use of the load information received.

The second type of Load report is a PEER report. This report is used by Diameter nodes as part of the logic to select the next-hop Diameter node and, as such, does not have significance beyond the peer node. Load reports of type PEER are removed by the first supporting Diameter node to receive the report.

Because Load reports can traverse Diameter nodes that do not support the Load mechanism, it is necessary to include the identity of the node to which the Load report applies as part of the Load report. This allows for a Diameter node to verify that a Load report applies to its peer or if it should be ignored.

The Load report includes a value indicating relative load of the sending node, specified in a manner consistent with that defined for DNS SRV [RFC2782].

The goal is to make it possible to use both the load values received as a part of the Diameter Load mechanism and weight values received as a result of a DNS SRV query. As a result, the Diameter load value has a range of 0-65535. This value and DNS SRV weight values are then used in a distribution algorithm similar to that specified in [RFC2782].

The DNS SRV distribution algorithm results in more messages being sent to a node with a higher weight value. As a result, a higher Diameter load value indicates a LOWER load on the sending node. A node that is heavily loaded sends a lower Diameter load value. Stated another way, a node that has zero load would have a load value of 65535. A node that is 100% loaded would have a load value of 0.

The distribution algorithm used by Diameter nodes supporting the Diameter Load mechanism is an implementation decision but it needs to result in similar behavior to the algorithm described for the use of weight values specified in [RFC2782].

The method for calculating the load value included in the Load report is also left as an implementation decision.

The frequency for sending of Load reports is also left as an implementation decision. The sending node might choose to send Load reports in all messages or it might choose to only send Load reports when the load value has changed by some implementation specific amount. The important consideration is that all nodes needing the load information have a sufficiently accurate view of the node's load.

5.1. Theory of Operation

This section outlines how the Diameter Load mechanism is expected to work.

For this discussion, assume the following Diameter network configuration:

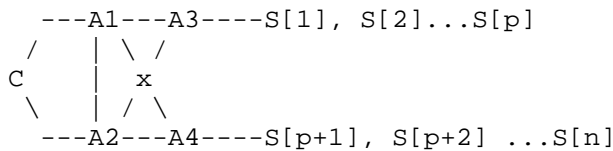


Figure 1: Example Diameter Network

Note that in this diagram, S[1], S[2] through S[p] are peers to A3. S[p+1], S[p+2] through S[n] are peers to A4.

Also assume that the request for a Diameter transaction takes the following path:

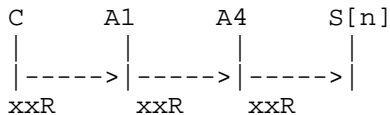


Figure 2: Request Message Path

When sending the answer message, an endpoint node that supports the Diameter Load mechanism includes its own load information in the answer message. Because it is a Diameter endpoint it includes a HOST Load report.

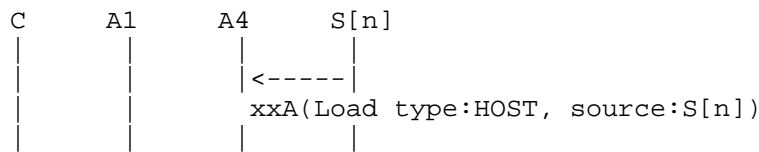


Figure 3: Answer Message from S[n]

If Agent A4 supports the Load mechanism then A4's actions depend on whether A4 is responsible for doing server selection. If A4 is not doing server selection then A4 ignores the HOST Load report. If A4 is responsible for doing server selection then it stores the load information for S[n] in its routing information for the handling of subsequent request messages. In both cases A4 leaves the HOST report in the message.

Note: If A4 does not support the Load mechanism then it will relay the answer message without doing any processing on the load information. In this case the load information AVPs will be relayed without change.

A4 then calculates its own load information and inserts load information AVPs of type PEER in the message before sending the message to A1.

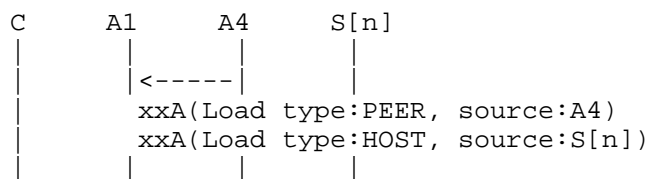


Figure 4: Answer Message from A4

If A1 supports the Load mechanism then it processes each of the Load reports it receives separately.

For the PEER Load report, A1 first determines if the source of the report indicated in the Load report matches the DiameterIdentity of the Diameter node from which the request was received. If the identities do not match then the PEER load report is discarded. If the identities match then A1 saves the load information in its routing information for routing of subsequent request messages. In both cases A1 strips the PEER Load report from the message.

For the HOST Load report, A1's actions depend on whether A1 is responsible for doing server selection. If A1 is not doing server selection then A1 ignores the HOST Load report. If A1 is responsible for doing server selection then it stores the load information for S[n] in its routing information for the handling of subsequent request messages. In both cases A1 leaves the HOST report in the message.

A1 then calculates its own load information and inserts load information AVPs of type PEER in the message before sending the message to C:

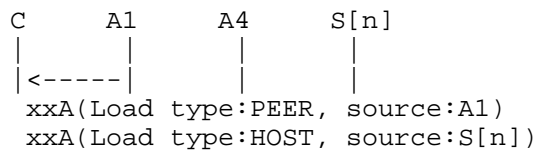


Figure 5: Answer Message from A1

As with A1, C processes each Load report separately.

For the PEER Load report, C follows the same procedure as A1 for determining if the Load report was received from the peer from which the report was sent. When finding it does, C stores the load information for use when making future routing decisions.

For the HOST Load report, C saves the load information only if it is responsible for doing server selection.

The Load information received by all nodes is then used for routing of subsequent request messages.

6. Load Mechanism Procedures

This section defines the normative behaviors for the Load mechanism.

6.1. Reporting Node Behavior

This section defines the procedures of Diameter reporting nodes that generate Load reports.

6.1.1. Endpoint Reporting Node Behavior

A Diameter endpoint that supports the Diameter Load mechanism MUST include a Load report of type HOST in sufficient answer messages to

ensure that all consumers of the load information receive timely updates.

The Diameter endpoint MUST include its own DiameterIdentity in the SourceID AVP included in the Load AVP.

The Diameter endpoint MUST include a Load-Type AVP of type HOST in the Load AVP.

The Diameter endpoint MUST include its load value in the Load-Value AVP in the Load AVP.

The LOAD value should be calculated in a way that reflects the available load independently of the weight of each server, in order to accurately compare LOAD values from different nodes. Any specific LOAD value needs to identify the same amount of available capacity, regardless the Diameter node that calculates the value.

The mechanism used to calculate the LOAD value that fulfills this requirement is an implementation decision.

The frequency of sending Load reports is an implementation decision.

For instance, if the only consumer of the Load reports is the endpoint's peer then the endpoint can choose to only include a Load report when the load of the endpoint has changed by a meaningful percentage. If there are consumers of the endpoint Load report other than the endpoint's peer (this will be the case if other nodes are responsible for server selection) then the endpoint might choose to include Load reports in all answer messages as a way of ensuring that all nodes doing server selection get accurate load information.

6.1.2. Agent Reporting Node Behavior

A Diameter Agent that supports the Diameter Load mechanism MUST include a PEER Load report in sufficient answer messages to ensure that all users of the load information receive timely updates.

The Diameter Agent MUST include its own DiameterIdentity in the SourceID AVP included in the Load AVP.

The Diameter Agent MUST include a Load-Type AVP of type PEER in the Load AVP.

The Diameter Agent MUST include its load value in the Load-Value AVP in the Load AVP.

The LOAD value should be calculated in a way that reflects the available load independently of the weight of each agent, in order to accurately compare LOAD values from different nodes. Any specific LOAD value needs to identify the same amount of available capacity, regardless the Diameter node that calculates the value.

The mechanism used to calculate the LOAD value that fulfills this requirement is an implementation decision.

The frequency of sending Load reports is an implementation decision.

Note: In the case of peer Load reports it is only necessary to include Load reports when the load value has changed by some meaningful value, as long as the agent ensures that all peers receive the report. It is also acceptable to include the Load report in every answer message handled by the Diameter Agent.

6.2. Reacting Node Behavior

This section defines the behavior of Diameter nodes processing Load reports.

A Diameter node MUST be prepared to process Load reports of type HOST and of type PEER, as indicated in the Load-Type AVP included in the Load AVP received in the same answer message or from multiple answer messages.

Note that the node needs to be able to handle messages with no load reports, messages with just a PEER Load report, messages with just an HOST Load report and messages with both types of Load reports.

If the Diameter node is not responsible for doing server selection then it SHOULD ignore Load reports of type HOST.

If the Diameter node is responsible for doing server selection then it SHOULD save the load value included in the Load-Value AVP included in the Load AVP of type HOST in its routing information.

If the Diameter node receives a Load report of type PEER then the Diameter node MUST determine if the Load report was inserted into the answer message by the peer from which the message was received. This is achieved by comparing the DiameterIdentity associated with the connection from which the message was received with the DiameterIdentity included in the SourceID AVP in the Load report.

If the Diameter node determines that the Load report of type PEER was not received from the peer that sent or relayed the answer message then the node MUST ignore the Load report.

If the Diameter node determines that the Load report of type PEER was received from the peer that sent or relayed the answer message then the node SHOULD save the load information in its routing information.

In all cases, a Diameter Agent MUST strip all Load reports of type PEER received in answer messages.

Note: This ensures that there will be precisely one Load report of type PEER, that of the Diameter node sending the message, in any answer messages sent by the Diameter Agent.

How a Diameter node uses load information for making routing decisions is an implementation decision. However, the distribution algorithm MUST result in similar behavior as the algorithm described for the use of weight values in [RFC2782].

6.3. Extensibility

The Load mechanism can be extended to include additional information in the Load reports.

Any extension may define new AVPs for use in Load reports. These new AVPs SHOULD be defined to be extensions to the Load AVPs defined in this document.

[RFC6733] defined Grouped AVP extension mechanisms apply. This allows, for example, defining a new feature that is mandatory to be understood even when piggybacked on an existing application.

As with any Diameter specification, [RFC6733] requires all new AVPs to be registered with IANA. See Section 9 for the required procedures.

6.4. Addition and Removal of Nodes

When a Diameter node is added, the new node will start by advertising its load. Downstream nodes will need to factor the new load information into load balancing decisions. The downstream nodes can attempt to ensure a smooth increase of the traffic to the new node, avoiding an immediate spike of traffic to the new node. The method for handling of such a smooth increase is implementation specific but it can rely on the evolution of load information received from the new node and from the other nodes.

When removing a node in a controlled way (e.g. for maintenance purpose, so outside a failure case), it might be appropriate to progressively reduce the traffic to this node by routing traffic to other nodes. Simple load information (load percentage) would not be sufficient. The method for handling of the node removal is implementation specific but it can rely on the evolution of the load information received from the node to be removed.

7. Attribute Value Pairs

The section defines the AVPs required for the Load mechanism.

7.1. Load AVP

The Load AVP (AVP code TBD1) is of type Grouped and is used to convey load information between Diameter nodes.

```
Load ::= < AVP Header: TBD1 >
        [ Load-Type ]
        [ Load-Value ]
        [ SourceID ]
        * [ AVP ]
```

7.2. Load-Type AVP

The Load-Type AVP (AVP code TBD2) is of type Enumerated. It is used to convey the type of Diameter node that sent the load information. The following values are defined:

HOST 0 The Load report is for a host.

PEER 1 The Load report is for a peer.

7.3. Load-Value AVP

The Load-Value AVP (AVP code TBD3) is of type Unsigned64. It is used to convey relative load information about the sender of the Load report.

The Load-Value AVP is specified in a manner similar to the weight value in DNS SRV ([RFC2782]).

The Load-Value has a range of 0-65535.

A higher value indicates a lower load on the sending node. A lower value indicates that the sending node is heavily loaded.

Stated another way, a node that has zero load would have a load value of 65535. A node that is 100% loaded would have a load value of 0.

7.4. SourceID AVP

The SourceID AVP is defined in [I-D.ietf-dime-agent-overload]. It is used to identify the Diameter node that sent the Load report.

7.5. Attribute Value Pair flag rules

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules	
				MUST	NOT
Load	TBD1	x.1	Grouped	V	
Load-Type	TBD2	x.2	Enumerated	V	
Load-Value	TBD3	x.3	Unsigned64	V	
SourceID	TBD4	x.4	DiameterIdentity	V	

As described in the Diameter base protocol [RFC6733], the M-bit usage for a given AVP in a given command may be defined by the application.

8. Security Considerations

Load information may be sensitive information in some cases. Depending on the mechanism, an unauthorized recipient might be able to infer the topology of a Diameter network from load information. Load information might be useful in identifying targets for Denial of Service (DoS) attacks, where a node known to be already heavily loaded might be a tempting target. Load information might also be useful as feedback about the success of an ongoing DoS attack.

Given that routing decisions are impacted by load information, there is potential for negative impacts on a Diameter network caused by erroneous or malicious load reports. This includes the malicious changing of load values by Diameter Agents.

Any load information conveyance mechanism will need to allow operators to avoid sending load information to nodes that are not

authorized to receive it. Since Diameter currently only offers authentication of nodes at the transport level, any solution that sends load information to non-peer nodes requires a transitive-trust model.

9. IANA Considerations

9.1. AVP Codes

New AVPs defined by this specification are listed in Section Section 7. All AVP codes are allocated from the 'Authentication, Authorization, and Accounting (AAA) Parameters' AVP Codes registry.

9.2. New Registries

This document makes no new registry requests of IANA.

10. References

10.1. Normative References

[I-D.ietf-dime-agent-overload]

Donovan, S., "Diameter Agent Overload", draft-ietf-dime-agent-overload-02 (work in progress), August 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <<http://www.rfc-editor.org/info/rfc2782>>.

[RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.

[RFC7683] Korhonen, J., Ed., Donovan, S., Ed., Campbell, B., and L. Morand, "Diameter Overload Indication Conveyance", RFC 7683, DOI 10.17487/RFC7683, October 2015, <<http://www.rfc-editor.org/info/rfc7683>>.

10.2. Informative References

- [RFC7068] McMurry, E. and B. Campbell, "Diameter Overload Control Requirements", RFC 7068, DOI 10.17487/RFC7068, November 2013, <<http://www.rfc-editor.org/info/rfc7068>>.

Appendix A. Topology Scenarios

This section presents a number of Diameter topology scenarios, and discusses how load information might be used in each scenario.

A.1. No Agent

Figure 6 shows a simple client-server scenario, where a client picks from a set of candidate servers available for a particular realm and application. The client selects the server for a given transaction using the load information received from each server.

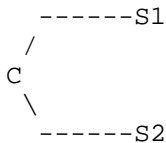


Figure 6: Basic Client Server Scenario

If a node supports dynamic discovery, it will not obtain load information from the nodes with which it has no Diameter connection established. Nevertheless it might take into account the load information from the other nodes to decide to add connections to new nodes with the dynamic discovery mechanism.

Note: The use of dynamic connections needs to be considered.

A.2. Single Agent

Figure 7 shows a client that sends requests to an agent. The agent selects the request destination from a set of candidate servers, using load information received from each server. The client does not need to receive load information, since it does not select between multiple agents.

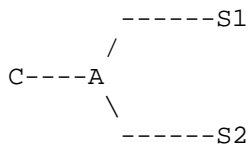


Figure 7: Simple Agent Scenario

A.3. Multiple Agents

Figure 8 shows a client selecting between multiple agents, and each agent selecting from multiple servers. The client selects an agent based on the load information received from each agent. Each agent selects a server based on the load information received from its servers.

This scenario adds a complication that one set of servers may be more loaded than the other set. If, for example, S4 was the least loaded server, C would need to know to select agent A2 to reach S4. This might require C to receive load information from the servers as well as the agents. Alternatively, each agent might use the load of its servers as an input into calculating its own load, in effect aggregating upstream load.

Similarly, if C sends a host-routed request [RFC7683], it needs to know which agent can deliver requests to the selected server. Without some special, potentially proprietary, knowledge of the topology upstream of A1 and A2, C would select the agent based on the normal peer selection procedures for the realm and application, and perhaps consider the load information from A1 and A2. If C sends a request to A1 that contains a Destination-Host AVP with a value of S4, A1 will not be able to deliver the request.

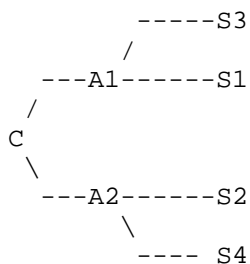


Figure 8: Multiple Agents and Servers

A.4. Linked Agents

Figure 9 shows a scenario similar to that of Figure 8, except that the agents are linked, so that A1 can forward a request to A2, and vice-versa. Each agent could receive load information from the linked agent, as well as its connected servers.

This somewhat simplifies the complication from Figure 8, due to the fact that C does not necessarily need to choose a particular agent to reach a particular server. But it creates a similar question of how, for example, A1 might know that S4 was less loaded than S1 or S3. Additionally, it creates the opportunity for sub-optimal request paths. For example [C,A1,A2,S4] vs. [C,A2,S4].

A likely application for linked agents is when each agent prefers to route only to directly connected servers and only forwards requests to another agent under exceptional circumstances. For example, A1 might not forward requests to A2 unless both S1 and S3 are overloaded. In this case, A1 might use the load information from S1 and S3 to select between those, and only consider the load information from A2 (and other connected agents) if it needs to divert requests to different agents.

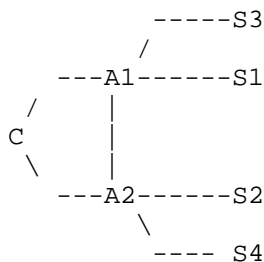


Figure 9: Linked Agents

Figure 10 is a variant of Figure 9. In this case, C1 sends all traffic through A1 and C2 sends all traffic through A2. By default, A1 will load balance traffic between S1 and S3 and A2 will load balance traffic between S2 and S4.

Now, if S1 S3 are significantly more loaded than S2 S4, A1 may route some C1 traffic to A2. This is non optimal path but allows a better load balancing between the servers. To achieve this, A1 needs to receive some load info from A2 about S2/S4 load.

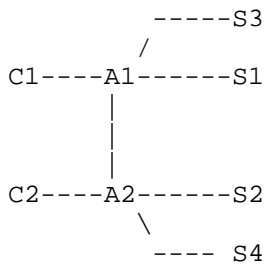


Figure 10: Linked Agents

A.5. Shared Server Pools

Figure 11 is similar to Figure 9, except that instead of a link between agents, each agent is linked to all servers. (The links to each set of servers should be interpreted as a link to each server. The links are not shown separately due to the limitations of ASCII art.)

In this scenario, each agent can select among all of the servers, based on the load information from the servers. The client need only be concerned with the load information of the agents.

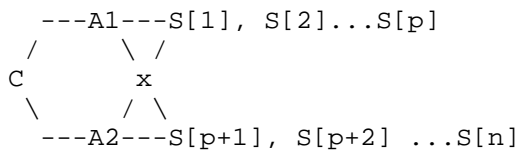


Figure 11: Shared Server Pools

A.6. Agent Chains

The scenario in Figure 12 is similar to that of Figure 8, except that, instead of the client possibly needing to select an agent that can route requests to the least loaded server, in this case A1 and A2 need to make similar decisions when selecting between A3 or A4. As the former scenario, this could be mitigated if A3 and A4 aggregate upstream loads into the load information they report downstream.

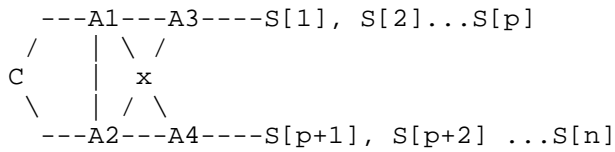


Figure 12: Agent Chains

A.7. Fully Meshed Layers

Figure 13 extends the scenario in Figure 11 by adding an extra layer of agents. But since each layer of nodes can reach any node in the next layer, each node only needs to consider the load of its next-hop peer.

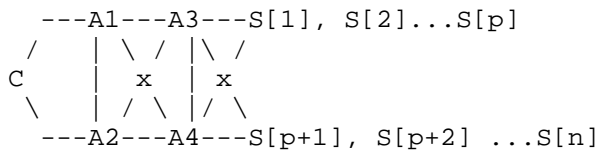


Figure 13: Full Mesh

A.8. Partitions

A Diameter network with multiple servers is said to be "partitioned" when only a subset of available servers can serve a particular realm-routed request. For example, one group of servers may handle users whose names start with "A" through "M", and another group may handle "N" through "Z".

In such a partitioned network, nodes cannot load-balance requests across partitions, since not all servers can handle the request. A client, or an intermediate agent, may still be able to load-balance between servers inside a partition.

A.9. Active-Standby Nodes

The previous scenarios assume that traffic can be load balanced among all peers that are eligible to handle a request. That is, the peers operate in an "active-active" configuration. In an "active-standby" configuration, traffic would be load-balanced among active peers. Requests would only be sent to peers in a "standby" state if the active peers became unavailable. For example, requests might be diverted to a stand-by peer if one or more active peers becomes overloaded.

Authors' Addresses

Ben Campbell
Oracle
7460 Warren Parkway # 300
Frisco, Texas 75034
USA

Email: ben@nostrum.com

Steve Donovan (editor)
Oracle
7460 Warren Parkway # 300
Frisco, Texas 75034
United States

Email: srdonovan@usdonovans.com

Jean-Jacques Trottin
Nokia
Route de Villejust
91620 Nozay
France

Email: jean-jacques.trottin@nokia.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2017

L. Bertz, Ed.
Sprint
D. Dolson, Ed.
Y. Lifshitz, Ed.
Sandvine
March 9, 2017

Diameter Credit-Control Application
draft-ietf-dime-rfc4006bis-02

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.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	5
1.1.	Requirements Language	6
1.2.	Terminology	6
1.3.	Advertising Application Support	8
2.	Architecture Models	8
3.	Credit-Control Messages	10
3.1.	Credit-Control-Request (CCR) Command	10
3.2.	Credit-Control-Answer (CCA) Command	11
4.	Credit-Control Application Overview	12
4.1.	Service-Specific Rating Input and Interoperability	14
4.1.1.	Specifying Rating Input AVPs	14
4.1.2.	Service-Specific Documentation	15
4.1.3.	Handling of Unsupported/Incorrect Rating Input	16
4.1.4.	RADIUS Vendor-Specific Rating Attributes	16
5.	Session Based Credit-Control	16
5.1.	General Principles	16
5.1.1.	Basic Tariff-Time Change Support	17
5.1.2.	Credit-Control for Multiple Services within a (sub-)Session	18
5.2.	First Interrogation	22
5.2.1.	First Interrogation after Authorization and Authentication	24
5.2.2.	Authorization Messages for First Interrogation	25
5.3.	Intermediate Interrogation	28
5.4.	Final Interrogation	30
5.5.	Server-Initiated Credit Re-Authorization	31
5.6.	Graceful Service Termination	33
5.6.1.	Terminate Action	36
5.6.2.	Redirect Action	37
5.6.3.	Restrict Access Action	39
5.6.4.	Usage of the Server-Initiated Credit Re-Authorization	40

5.7.	Failure Procedures	40
6.	One Time Event	43
6.1.	Service Price Enquiry	44
6.2.	Balance Check	45
6.3.	Direct Debiting	45
6.4.	Refund	46
6.5.	Failure Procedure	47
7.	Credit-Control Application State Machine	49
8.	Credit-Control AVPs	57
8.1.	CC-Correlation-Id AVP	60
8.2.	CC-Request-Number AVP	60
8.3.	CC-Request-Type AVP	60
8.4.	CC-Session-Failover AVP	61
8.5.	CC-Sub-Session-Id AVP	61
8.6.	Check-Balance-Result AVP	62
8.7.	Cost-Information AVP	62
8.8.	Unit-Value AVP	63
8.9.	Exponent AVP	63
8.10.	Value-Digits AVP	63
8.11.	Currency-Code AVP	63
8.12.	Cost-Unit AVP	64
8.13.	Credit-Control AVP	64
8.14.	Credit-Control-Failure-Handling AVP	64
8.15.	Direct-Debiting-Failure-Handling AVP	65
8.16.	Multiple-Services-Credit-Control AVP	66
8.17.	Granted-Service-Unit AVP	67
8.18.	Requested-Service-Unit AVP	68
8.19.	Used-Service-Unit AVP	68
8.20.	Tariff-Time-Change AVP	69
8.21.	CC-Time AVP	69
8.22.	CC-Money AVP	69
8.23.	CC-Total-Octets AVP	70
8.24.	CC-Input-Octets AVP	70
8.25.	CC-Output-Octets AVP	70
8.26.	CC-Service-Specific-Units AVP	70
8.27.	Tariff-Change-Usage AVP	70
8.28.	Service-Identifier AVP	71
8.29.	Rating-Group AVP	71
8.30.	G-S-U-Pool-Reference AVP	71
8.31.	G-S-U-Pool-Identifier AVP	72
8.32.	CC-Unit-Type AVP	72
8.33.	Validity-Time AVP	72
8.34.	Final-Unit-Indication AVP	73
8.35.	Final-Unit-Action AVP	74
8.36.	Restriction-Filter-Rule AVP	75
8.37.	Redirect-Server AVP	75
8.38.	Redirect-Address-Type AVP	76
8.39.	Redirect-Server-Address AVP	76

8.40.	Multiple-Services-Indicator AVP	76
8.41.	Requested-Action AVP	77
8.42.	Service-Context-Id AVP	78
8.43.	Service-Parameter-Info AVP	78
8.44.	Service-Parameter-Type AVP	79
8.45.	Service-Parameter-Value AVP	79
8.46.	Subscription-Id AVP	79
8.47.	Subscription-Id-Type AVP	80
8.48.	Subscription-Id-Data AVP	80
8.49.	User-Equipment-Info AVP	81
8.50.	User-Equipment-Info-Type AVP	81
8.51.	User-Equipment-Info-Value AVP	82
8.52.	User-Equipment-Info-Extension AVP	82
8.53.	User-Equipment-Info-IMEISV AVP	82
8.54.	User-Equipment-Info-MAC AVP	82
8.55.	User-Equipment-Info-EUI64 AVP	82
8.56.	User-Equipment-Info-ModifiedEUI64 AVP	83
8.57.	User-Equipment-Info-IMEI AVP	83
8.58.	Subscription-Id-Extension AVP	83
8.59.	Subscription-Id-E164 AVP	84
8.60.	Subscription-Id-IMSI AVP	84
8.61.	Subscription-Id-SIP-URI AVP	84
8.62.	Subscription-Id-NAI AVP	84
8.63.	Subscription-Id-Private AVP	84
8.64.	Redirect-Server-Extension AVP	84
8.65.	Redirect-Address-IPAddress AVP	85
8.66.	Redirect-Address-URL AVP	85
8.67.	Redirect-Address-SIP-URI AVP	85
8.68.	QoS-Final-Unit-Indication AVP	86
9.	Result Code AVP Values	87
9.1.	Transient Failures	87
9.2.	Permanent Failures	88
10.	AVP Occurrence Table	88
10.1.	Credit-Control AVP Table	89
10.2.	Re-Auth-Request/Answer AVP Table	90
11.	RADIUS/Diameter Credit-Control Interworking Model	90
12.	IANA Considerations	93
12.1.	Application Identifier	94
12.2.	Command Codes	94
12.3.	AVP Codes	94
12.4.	Result-Code AVP Values	94
12.5.	CC-Request-Type AVP	94
12.6.	CC-Session-Failover AVP	94
12.7.	CC-Unit-Type AVP	94
12.8.	Check-Balance-Result AVP	95
12.9.	Credit-Control AVP	95
12.10.	Credit-Control-Failure-Handling AVP	95
12.11.	Direct-Debiting-Failure-Handling AVP	95

12.12. Final-Unit-Action AVP	95
12.13. Multiple-Services-Indicator AVP	95
12.14. Redirect-Address-Type AVP	95
12.15. Requested-Action AVP	96
12.16. Subscription-Id-Type AVP	96
12.17. Tariff-Change-Usage AVP	96
12.18. User-Equipment-Info-Type AVP	96
13. Credit-Control Application Related Parameters	96
14. Security Considerations	97
14.1. Direct Connection with Redirects	98
15. References	98
15.1. Normative References	98
15.2. Informative References	101
Appendix A. Acknowledgements	101
Appendix B. Credit-Control Sequences	101
B.1. Flow I	101
B.2. Flow II	104
B.3. Flow III	106
B.4. Flow IV	106
B.5. Flow V	108
B.6. Flow VI	109
B.7. Flow VII	110
B.8. Flow VIII	112
B.9. Flow IX	114
Appendix C. Changes relative to RFC4006	119
Authors' Addresses	120

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 [TGPPCHARG] 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 [RFC2119].

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., [RFC7155] and [RFC4004].

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., [RFC7155] and [RFC4004].

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 [RFC6733].

2. Architecture Models

The current accounting models specified in the Radius Accounting [RFC2866] and Diameter base [RFC6733] 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, [RFC7155] and [RFC4004], 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.

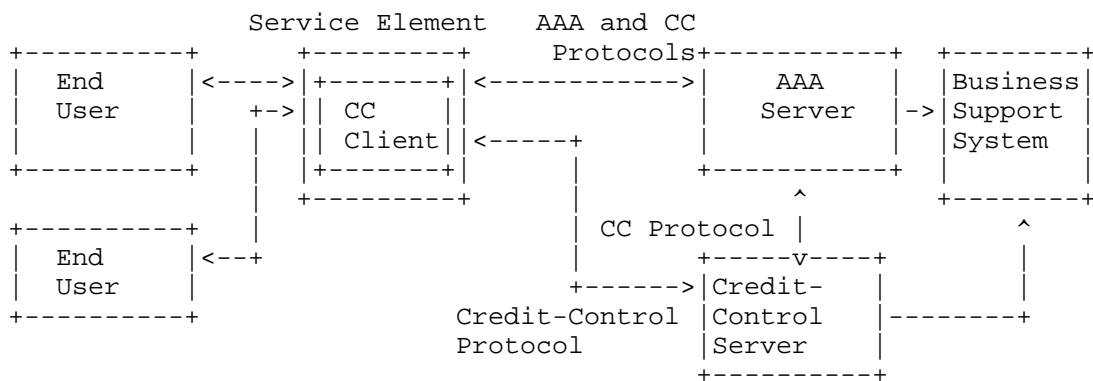


Figure 1: Typical credit-control architecture

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 [RFC6733], 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

Table 1: Credit-Control Commands

Diameter Base [RFC6733] defines in the section 3.2 the Command Code format 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 ]
        *[ Subscription-Id-Extension ]
        [ 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 ]
        [ User-Equipment-Info-Extension ]
        *[ 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 ]
    [ QoS-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., [RFC7155] for network access services). Re-use of existing AVPs is strongly recommended in [RFC6733].

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 [RFC6733], section 1.3.

1b. New AVPs can be defined if the existing AVPs do not provide sufficient rating information. In this case, the procedures defined in [RFC6733] 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 [RFC6733].

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.

4.1.2. 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, or 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 [RFC6733].

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 [RFC7155] 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 Figure 3 and Figure 4.

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 [RFC6733], 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.

5.1.1.1. 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.

5.1.1.2. 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 or QoS-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.

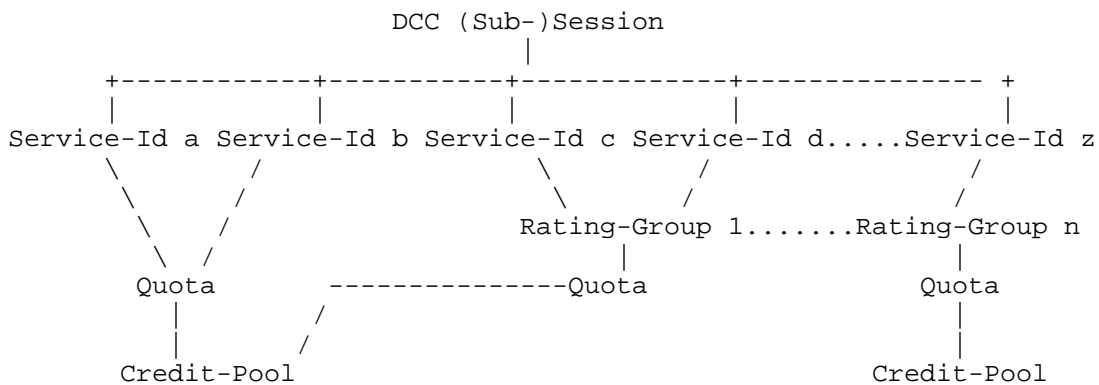


Figure 2: Multiple-Service (sub)-Session Example

If independent credit-control of multiple services is used, the Validity-Time AVP and Final-Unit-Indication AVP or QoS-Final-Unit-Indication AVP 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 AVP on the command level indicates a value other than SUCCESS, then

the Result-Code AVP 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:

$$S = Q_1 * M_1 + Q_2 * M_2 + \dots + Q_n * M_n$$

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 or the Subscription-Id-Extension 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 or User-Equipment-Info-Extension 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 or the QoS-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-control 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.

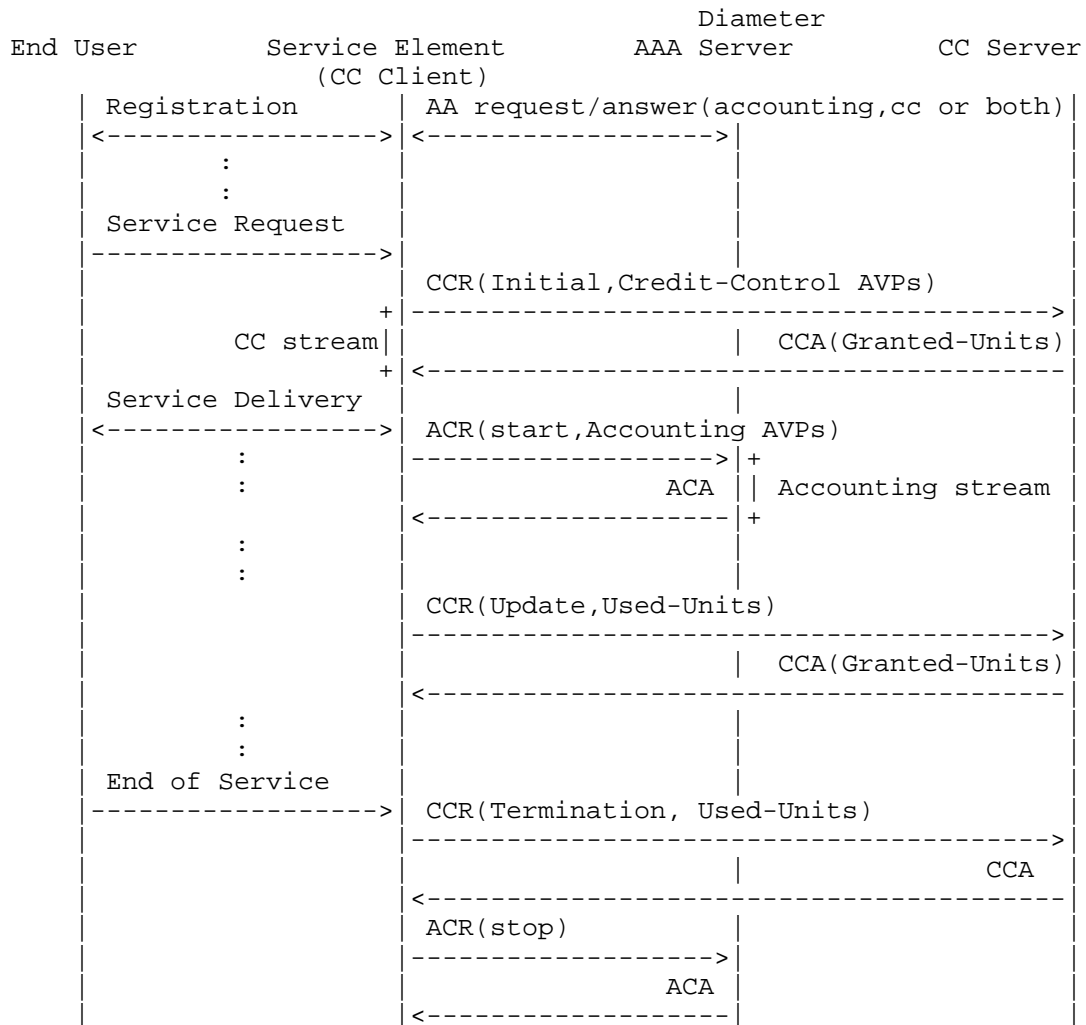


Figure 3: Protocol example with first interrogation after user's authorization/authentication

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., [RFC7155], [RFC4004]). 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., [RFC7155], [RFC4004]). 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.

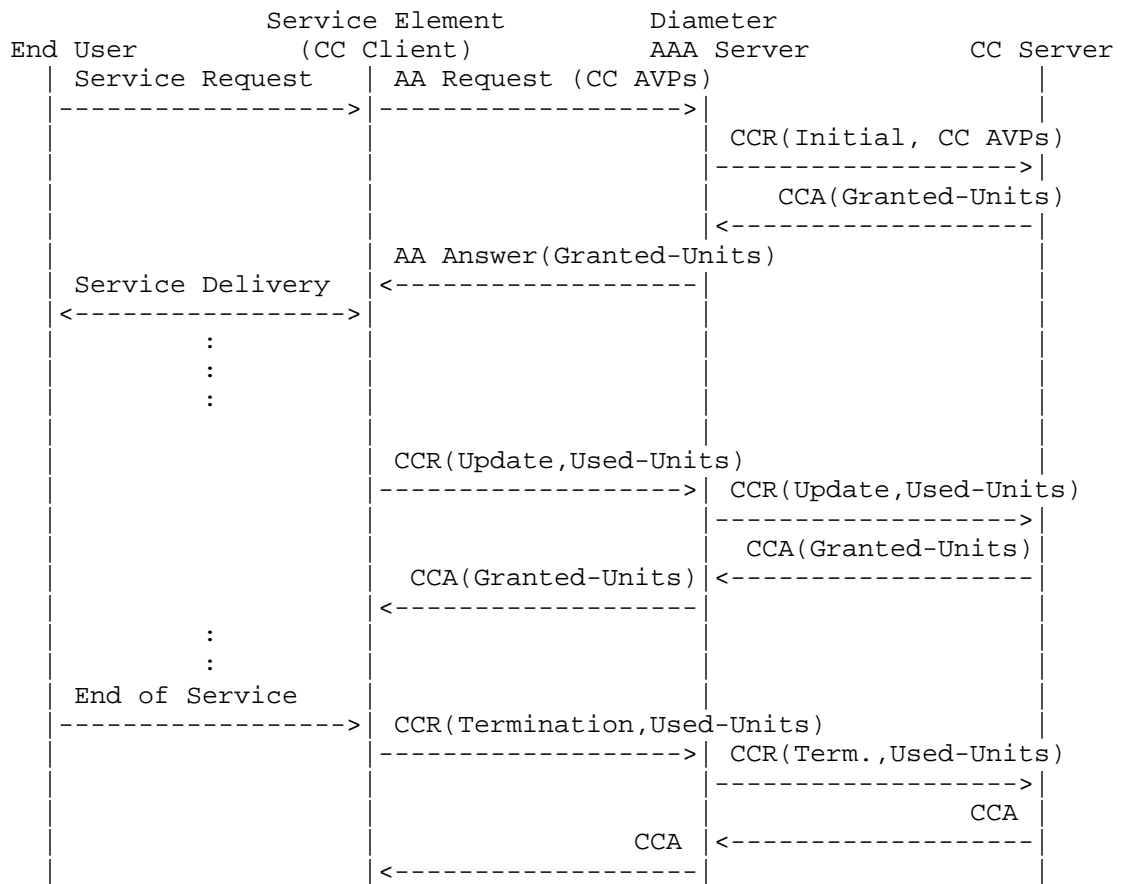


Figure 4: Protocol example with use of the authorization messages for the first interrogation

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 or Subscription-Id-Extension 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 or the QoS-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 [RFC6733]. Figure 5 illustrates the case when the final-unit

indication causes user logoff upon consumption of the final granted units and the generation of STR.

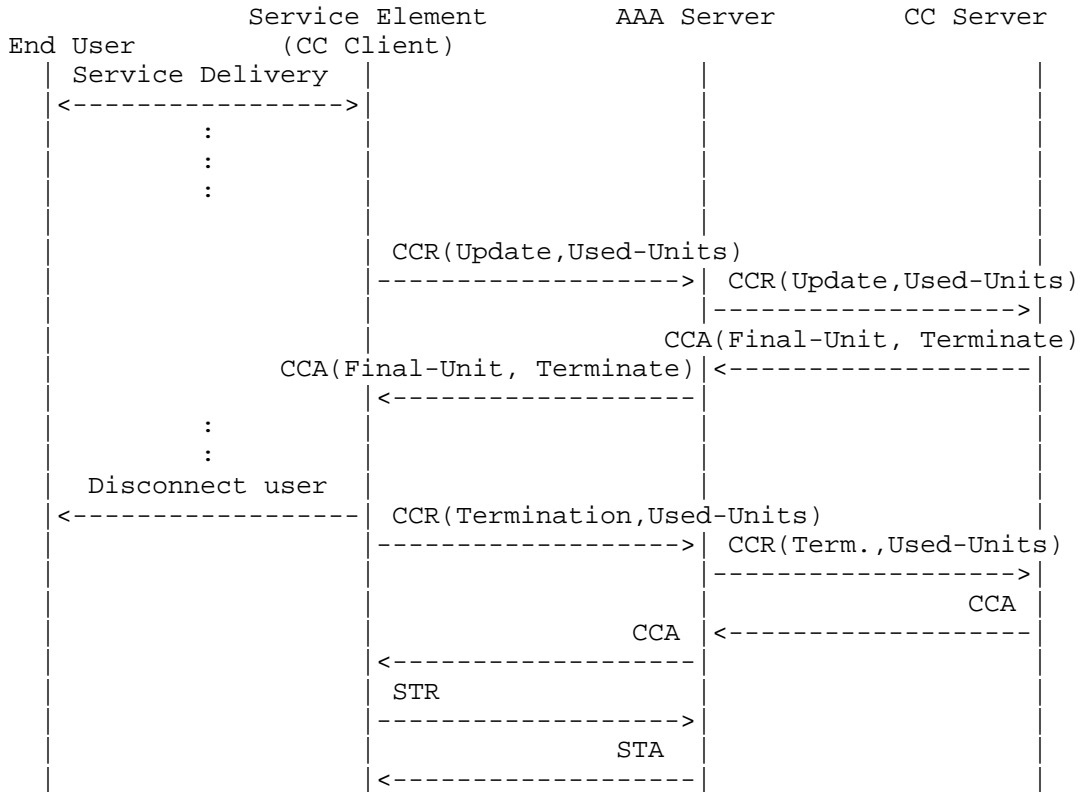


Figure 5: User disconnected due to exhausted account

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 [RFC6733]. 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 AVP or QoS-Final-Unit-Indication AVP 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 Appendix B.7.

The credit-control server MAY initiate the graceful service termination by including the Final-Unit-Indication AVP or the QoS-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 or the QoS-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.

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

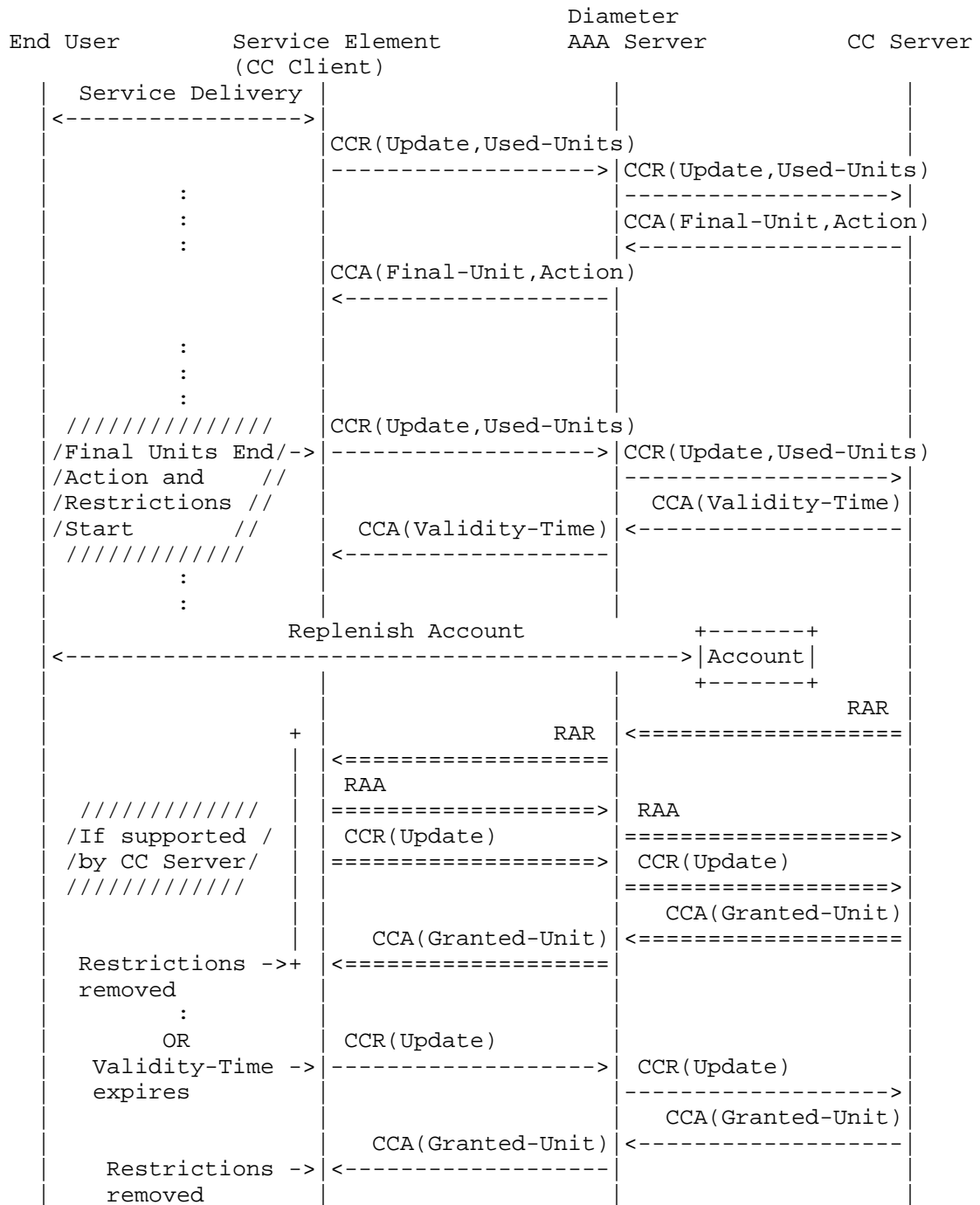


Figure 6: Optional graceful service termination procedure

In addition, the credit-control server MAY reply with Final-Unit-Indication AVP or QoS-Final-Unit-Indication AVP holding a G-S-U AVP with a zero grant, indicating that the service SHOULD be terminated immediately, and no further reporting is required. A following figure illustrates a graceful service termination procedure that applies immediately after receiving a zero grant.

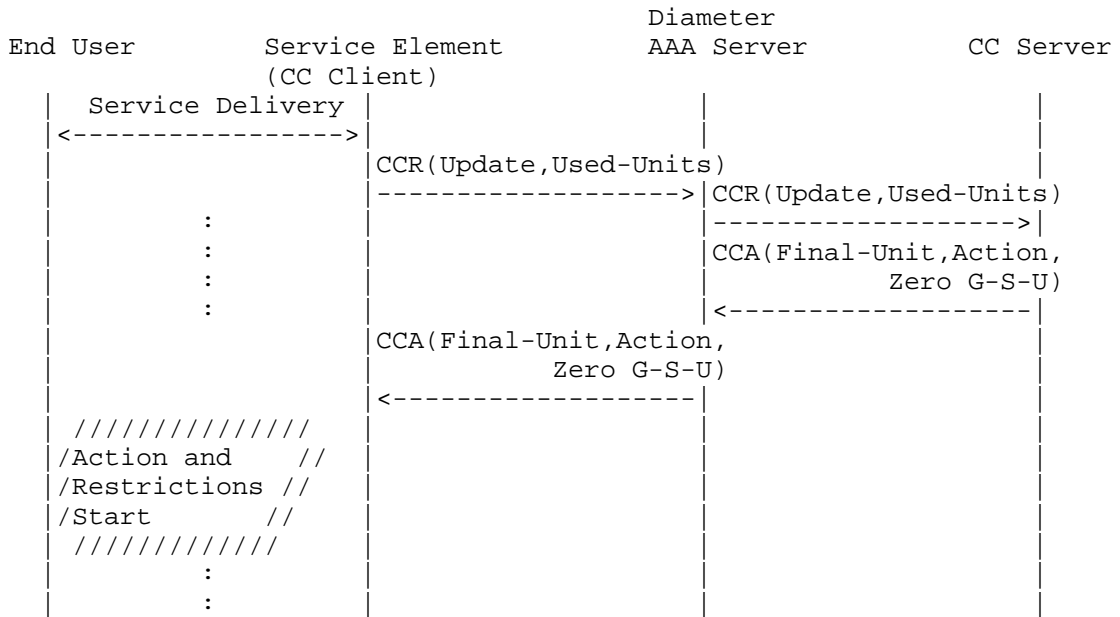


Figure 7: Optional immediate graceful service termination procedure

5.6.1. Terminate Action

The Final-Unit-Indication AVP or the QoS-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 or the QoS-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 or the QoS-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 or Redirect-Server-Extension AVP as follows.

The credit-control server sends the Redirect-Server AVP or Redirect-Server-Extension 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 or Redirect-Server-Extension 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 or Redirect-Server-Extension AVP, the credit-control server MAY include one or more Restriction-Filter-Rule AVPs, one or more 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 Restriction-Filter-Rule AVPs, Filter-Rule AVPs or Filter-Id AVPs. If enforcement actions other than allowing the packets (e.g., QoS), are indicated in the Filter-Rule AVPs or Filter-Id AVPs, they SHOULD be performed as well. In addition, if possible, to redirecting the user to the destination specified in the Redirect-Server AVP or Redirect-Server-Extension 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 B.8 (Figure 18).

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 or the QoS-Final-Unit-Indication AVP and Validity-Time AVPs but no Granted-Service-Unit AVP. It immediately starts the graceful service termination without sending any message to the server. An example of this case is illustrated in Appendix B.

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, upon consumption of the final granted units, 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 Final-Unit-Indication group AVP of the Credit-Control-Answer message.

A QoS-Final-Unit-Indication AVP with the Final-Unit-Action RESTRICT_ACCESS indicates to the device supporting this action that, upon consumption of the final granted units, the actions specified in Filter-Rule AVP(s) MUST restrict the traffic according to the classifiers in the Filter-Rule AVP(s). If Filter-Id AVP(s) are provided in the Credit-Control-Answer message, the credit control client MUST restrict the traffic according to the IP packet filters identified by the Filter-Id AVP(s). The credit-control server SHOULD include either the Filter-Rule AVP or the Filter-Id AVP in the QoS-Final-Unit-Indication group AVP of the Credit-Control-Answer message.

If both Final-Unit-Indication AVP and QoS-Final-Unit-Indication AVP exist in the Credit-Control-Answer message, a credit control client which supports the QoS-Final-Unit-Indication AVP SHOULD follow the directives included in the QoS-Final-Unit-Indication AVP and SHOULD ignore the Final-Unit-Indication 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. 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-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 [RFC6733]. 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 an 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 [RFC6733]. 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 [RFC3539] 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 [RFC3539]. The recommended default initial value for Tw (Twinit) is 30 seconds. Twinit may be set as low as 6 seconds; however, according to [RFC3539], 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 failback 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 8.

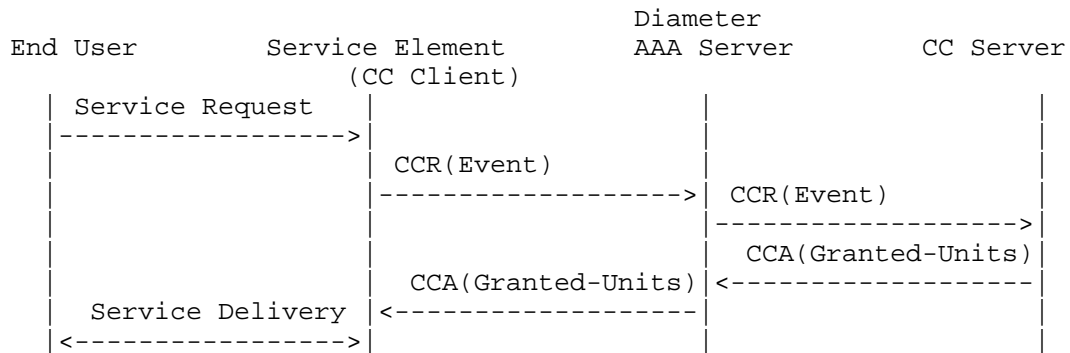


Figure 8: One time event

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 services 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 or Subscription-Id-Extension 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 or Subscription-Id-Extension 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 or Subscription-Id-Extension 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 [RFC6733]. 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 [RFC6733], 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 [RFC6733], 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 [RFC6733], 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 [RFC6733].

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 [RFC6733].

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.

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

Table 2: CLIENT, SESSION BASED for the first interrogation with AA request

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

Table 3: CLIENT, SESSION BASED for the first interrogation with CCR

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

Table 4: CLIENT, SESSION BASED for intermediate and final interrogations

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	Indicate service	Idle

	received, or Tx expired; requested action CHECK_BALANCE or PRICE_ENQUIRY	error	
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, 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

Table 5: CLIENT, 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

Table 6: SERVER, SESSION AND EVENT BASED

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 [RFC7155] and [RFC4004], 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 [RFC6733], 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, and possible flag values. The AVP Flag rules are explained in the Diameter base [RFC6733], section 4.1.

Attribute Name	AVP Code	Section Defined	Data Type	AVP Flag rules		
				MUST	MAY	MUST NOT
CC-Correlation-Id	411	8.1	OctetString		M	V
CC-Input-Octets	412	8.24	Unsigned64	M		V
CC-Money	413	8.22	Grouped	M		V
CC-Output-Octets	414	8.25	Unsigned64	M		V
CC-Request-Number	415	8.2	Unsigned32	M		V
CC-Request-Type	416	8.3	Enumerated	M		V
CC-Service-Specific-Units	417	8.26	Unsigned64	M		V
CC-Session-Failover	418	8.4	Enumerated	M		V
CC-Sub-Session-Id	419	8.5	Unsigned64	M		V
CC-Time	420	8.21	Unsigned32	M		V
CC-Total-Octets	421	8.23	Unsigned64	M		V
CC-Unit-Type	454	8.32	Enumerated	M		V
Check-Balance-Result	422	8.6	Enumerated	M		V
Cost-Information	423	8.7	Grouped	M		V
Cost-Unit	424	8.12	UTF8String	M		V
Credit-Control	426	8.13	Enumerated	M		V
Credit-Control-Failure-Handling	427	8.14	Enumerated	M		V

Currency-Code	425	8.11	Unsigned32	M		V
Direct-Debiting- Failure-Handling	428	8.15	Enumerated	M		V
Exponent	429	8.9	Integer32	M		V
Final-Unit-Action	449	8.35	Enumerated	M		V
Final-Unit- Indication	430	8.34	Grouped	M		V
QoS-Final-Unit- Indication	TBD17	8.68	Grouped		M	V
Granted-Service- Unit	431	8.17	Grouped	M		V
G-S-U-Pool- Identifier	453	8.31	Unsigned32	M		V
G-S-U-Pool- Reference	457	8.30	Grouped	M		V
Multiple-Services -Credit-Control	456	8.16	Grouped	M		V
Multiple-Services -Indicator	455	8.40	Enumerated	M		V
Rating-Group	432	8.29	Unsigned32	M		V
Redirect-Address -Type	433	8.38	Enumerated	M		V
Redirect-Server	434	8.37	Grouped	M		V
Redirect-Server -Address	435	8.39	UTF8String	M		V
Redirect-Server -Extension	TBD13	8.64	Grouped		M	V
Redirect-Address -IPAddress	TBD14	8.65	Address		M	V
Redirect-Address -URL	TBD15	8.66	UTF8String		M	V
Redirect-Address -SIP-URI	TBD16	8.67	UTF8String		M	V
Requested-Action	436	8.41	Enumerated	M		V
Requested-Service -Unit	437	8.18	Grouped	M		V
Restriction -Filter-Rule	438	8.36	IPFilterRule	M		V
Service-Context -Id	461	8.42	UTF8String	M		V
Service- Identifier	439	8.28	Unsigned32	M		V
Service-Parameter -Info	440	8.43	Grouped		M	V
Service- Parameter-Type	441	8.44	Unsigned32		M	V
Service- Parameter-Value	442	8.45	OctetString		M	V

Subscription-Id	443	8.46	Grouped	M		V
Subscription-Id	444	8.48	UTF8String	M		V
-Data						
Subscription-Id	450	8.47	Enumerated	M		V
-Type						
Subscription-Id	TBD7	8.58	Grouped		M	V
-Extension						
Subscription-Id	TBD8	8.59	UTF8String		M	V
-E164						
Subscription-Id	TBD9	8.60	UTF8String		M	V
-IMSI						
Subscription-Id	TBD10	8.61	UTF8String		M	V
-SIP-URI						
Subscription-Id	TBD11	8.62	UTF8String		M	V
-NAI						
Subscription-Id	TBD12	8.63	UTF8String		M	V
-Private						
Tariff-Change	452	8.27	Enumerated	M		V
-Usage						
Tariff-Time	451	8.20	Time	M		V
-Change						
Unit-Value	445	8.8	Grouped	M		V
Used-Service-Unit	446	8.19	Grouped	M		V
User-Equipment	458	8.49	Grouped		M	V
-Info						
User-Equipment	459	8.50	Enumerated		M	V
-Info-Type						
User-Equipment	460	8.51	OctetString		M	V
-Info-Value						
User-Equipment	TBD1	8.52	Grouped		M	V
-Info-Extension						
User-Equipment	TBD2	8.53	OctetString		M	V
-Info-IMEISV						
User-Equipment	TBD3	8.54	OctetString		M	V
-Info-MAC						
User-Equipment	TBD4	8.55	OctetString		M	V
-Info-EUI64						
User-Equipment	TBD5	8.56	OctetString		M	V
-Info-ModifiedEUI64						
User-Equipment	TBD6	8.57	OctetString		M	V
-Info-IMEI						
Value-Digits	447	8.10	Integer64	M		V
Validity-Time	448	8.33	Unsigned32	M		V

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** 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 an 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 [RFC6733]):

```
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., Unit-Value = Value-Digits AVP * 10^{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 [RFC6733]):

```
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 [RFC6733] 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 or QoS-Final-Unit-Indication AVPs MAY be present in an answer command as defined in Section 5.1.2 and Section 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 [RFC6733]):

```
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 ]
    [ QoS-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 [RFC6733]):

```
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 [RFC6733]):

```
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.

Note: The values reported in a Used-Service-Unit AVP does not necessarily have a relation to the grant provided in a Granted-Service-Unit AVP, e.g., the value in this AVP may exceed the value in the grant.

The Used-Service-Unit AVP is defined as follows (per the grouped-avp-def of [RFC6733]):

```
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 [RFC6733]):

```
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 B.9.

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 B.9.

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 [RFC6733]):

```
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, the Final-Unit-Indication group MUST NOT contain any other AVPs.

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 [RFC7155]. 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.

If the Final-Unit-Action AVP is set to REDIRECT and the type of server is not one of the enumerations in the Redirect-Address-Type AVP, then the QoS-Final-Unit-Indication AVP SHOULD be used together with the Redirect-Server-Extension AVP instead of the Final-Unit-Indication AVP.

If the Final-Unit-Action AVP is set to RESTRICT_ACCESS or REDIRECT and the classification of the restricted traffic cannot be expressed using IPFilterRule, or different actions (e.g., QoS) than just allowing QoS needs to be enforced traffic, then the QoS-Final-Unit-Indication AVP SHOULD be used instead of the Final-Unit-Indication AVP. However, if the credit control server wants to preserve backward compatibility with credit-control clients that support only [RFC4006], the Final-Unit-Indication AVP SHOULD be used together with the Filter-Id AVP.

The Final-Unit-Indication AVP is defined as follows (per the grouped-avp-def of [RFC6733]):

```
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 or one of the AVPs included in the Redirect-Server-Extension 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 filter AVPs contained in the applied grouped AVP: according to IP packet filters defined in the Restriction-Filter-Rule AVP, according to the packet classifier filters defined in Filter-Rule AVP, or according to the packet filters identified by the Filter-Id AVP. All the packets not matching any 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 [RFC6733]):

```
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 [RFC0791].

IPv6 Address 1

The address type is in the form of IPv6 address, as defined in [RFC4291]. The address MUST conform to the text representation of the address according to [RFC5952].

URL 2

The address type is in the form of Uniform Resource Locator, as defined in [RFC1738].

SIP URI 3

The address type is in the form of SIP Uniform Resource Identifier, as defined in [RFC3261].

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 [RFC6733]):

```
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 [RFC6733]):

```
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 [RFC6733].

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 [RFC3261].

END_USER_NAI 3

The identifier is in the form of a Network Access Identifier, as defined in [RFC2486].

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 [RFC6733]):

```
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 [TGPPIMEI].

MAC 1

The 48-bit MAC address is formatted as described in [RFC3580].

EUI64 2

The 64-bit identifier used to identify the 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 [RFC4291] 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.

8.52. User-Equipment-Info-Extension AVP

The User-Equipment-Info-Extension AVP (AVP Code TBD1) 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. If the type of the equipment is one of the enumerated types of User-Equipment-Info-Type AVP, then the credit-control client SHOULD send the information in the User-Equipment-Info AVP, in addition to or instead of the User-Equipment-Info-Extension AVP. This is in order to preserve backward compatibility with credit-control servers that support only RFC4006. Exactly one AVP MUST be included inside the User-Equipment-Info-Extension AVP.

It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
User-Equipment-Info-Extension ::= < AVP Header: TBD1 >
    [ User-Equipment-Info-IMEISV ]
    [ User-Equipment-Info-MAC ]
    [ User-Equipment-Info-EUI64 ]
    [ User-Equipment-Info-ModifiedEUI64 ]
    [ User-Equipment-Info-IMEI ]
    [ AVP ]
```

8.53. User-Equipment-Info-IMEISV AVP

The User-Equipment-Info-IMEISV (AVP Code TBD2) is of type OctetString. The User-Equipment-Info-IMEISV AVP contains the International Mobile Equipment Identifier and Software Version in the international IMEISV format according to 3GPP TS 23.003 [TGPPIMEI].

8.54. User-Equipment-Info-MAC AVP

The User-Equipment-Info-MAC (AVP Code TBD3) is of type OctetString. The User-Equipment-Info-MAC AVP contains the 48-bit MAC address is formatted as described in [RFC3580].

8.55. User-Equipment-Info-EUI64 AVP

The User-Equipment-Info-EUI64 (AVP Code TBD4) is of type OctetString. The User-Equipment-Info-EUI64 AVP contains the 64-bit identifier

used to identify the hardware instance of the product, as defined in [EUI64].

8.56. User-Equipment-Info-ModifiedEUI64 AVP

The User-Equipment-Info-ModifiedEUI64 (AVP Code TBD5) is of type OctetString. 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 [RFC4291] or by using some other methods referred to in the service-specific documentation. The User-Equipment-Info-ModifiedEUI64 AVP contains such identifiers.

8.57. User-Equipment-Info-IMEI AVP

The User-Equipment-Info-IMEI (AVP Code TBD6) is of type OctetString. The User-Equipment-Info-IMEI AVP contains the International Mobile Equipment Identifier in the international IMEI format according to 3GPP TS 23.003 [TGPPIMEI].

8.58. Subscription-Id-Extension AVP

The Subscription-Id-Extension AVP (AVP Code TBD7) is used to identify the end user's subscription and is of type Grouped. The Subscription-Id-Extension group AVP MUST include an AVP holding the subscription identifier. The type of this included AVP indicates the type of the subscription identifier. For each of the enumerated values of the Subscription-Id-Type AVP, there is a corresponding sub-AVP for use within the Subscription-Id-Extension group AVP. If a new identifier type is required a corresponding new sub-AVP SHOULD be defined for use within the Subscription-Id-Extension group AVP.

If full backward compatibility with [RFC4006] is required, then the Subscription-Id AVP MUST be used to indicate identifier types enumerated in the Subscription-Id-Type AVP, whereas the Subscription-Id-Extension AVP MUST be used only for newly defined identifier types. If full backward compatibility with [RFC4006] is not required, then the Subscription-Id-Extension AVP MAY be used to carry out the existing identifier types. In this case, Subscription-Id-Extension AVP MAY be sent together with Subscription-Id AVP.

Exactly one sub-AVP MUST be included inside the Subscription-Id-Extension AVP.

It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Subscription-Id-Extension ::= < AVP Header: TBD7 >
    [ Subscription-Id-E164 ]
    [ Subscription-Id-IMSI ]
    [ Subscription-Id-SIP-URI ]
    [ Subscription-Id-NAI ]
    [ Subscription-Id-Private ]
    [ AVP ]
```

8.59. Subscription-Id-E164 AVP

The Subscription-Id-E164 (AVP Code TBD8) is of type UTF8String. The Subscription-Id-E164 AVP contains the international E.164 format (e.g., MSISDN), according to the ITU-T E.164 numbering plan defined in [E164] and [CE164].

8.60. Subscription-Id-IMSI AVP

The Subscription-Id-IMSI (AVP Code TBD9) is of type UTF8String. The Subscription-Id-IMSI AVP contains the international IMSI format, according to the ITU-T E.212 numbering plan as defined in [E212] and [CE212].

8.61. Subscription-Id-SIP-URI AVP

The Subscription-Id-SIP-URI (AVP Code TBD10) is of type UTF8String. The Subscription-Id-SIP-URI AVP contains the identifier in the form of a SIP URI, as defined in [RFC3261].

8.62. Subscription-Id-NAI AVP

The Subscription-Id-NAI (AVP Code TBD11) is of type UTF8String. The Subscription-Id-NAI AVP contains the identifier in the form of a Network Access Identifier, as defined in [RFC2486].

8.63. Subscription-Id-Private AVP

The Subscription-Id-Private (AVP Code TBD12) is of type UTF8String. The Subscription-Id-Private AVP contains a credit-control server private identifier.

8.64. Redirect-Server-Extension AVP

The Redirect-Server-Extension AVP (AVP Code TBD13) 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 inside the QoS-Final-Unit-Indication AVP when the Final-Unit-

Action AVP is set to REDIRECT. If the type of the redirect server is one of the enumerated values of the Redirect-Address-Type AVP, then the credit-control server SHOULD send the information in the Redirect-Server AVP, in addition to or instead of the Redirect-Server-Extension AVP. This is in order to preserve backward compatibility with credit-control clients that support only [RFC4006]. Exactly one AVP MUST be included inside the Redirect-Server-Extension AVP.

It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Redirect-Server-Extension ::= < AVP Header: TBD13 >
    [ Redirect-Address-IPAddress ]
    [ Redirect-Address-URL ]
    [ Redirect-Address-SIP-URI ]
    [ AVP ]
```

8.65. Redirect-Address-IPAddress AVP

The Redirect-Address-IPAddress AVP (AVP Code TBD14) is of type Address and defines the IPv4 or IPv6 address of the redirect server with which the end user is to be connected when the account cannot cover the service cost.

When encoded as an IPv6 address in 16 bytes, the IPv4-mapped IPv6 format [RFC4291] MAY be used to indicate an IPv4 address.

8.66. Redirect-Address-URL AVP

The Redirect-Address-URL AVP (AVP Code TBD15) is of type UTF8String and defines the address of the redirect server with which the end user is to be connected when the account cannot cover the service cost. The address type is in the form of Uniform Resource Locator, as defined in [RFC1738].

8.67. Redirect-Address-SIP-URI AVP

The Redirect-Address-SIP-URI AVP (AVP Code TBD16) is of type UTF8String and defines the address of the redirect server with which the end user is to be connected when the account cannot cover the service cost. The address type is in the form of SIP Uniform Resource Identifier, as defined in [RFC3261].

8.68. QoS-Final-Unit-Indication AVP

The QoS-Final-Unit-Indication AVP (AVP Code TBD17) 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 QoS-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 QoS-Final-Unit-Indication AVP is included in a command.

If the Final-Unit-Action AVP is set to TERMINATE, the QoS-Final-Unit-Indication group MUST NOT contain any other AVPs.

If the Final-Unit-Action AVP is set to REDIRECT at least the Redirect-Server-Extension AVP MUST be present. The 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-Extension AVP or if the access to these services needs to be limited in some way (e.g., QoS).

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

The Filter-Rule AVP is defined in [RFC5777]. The Filter-Rule AVP can be used to define a specific condition and action combination. If used only with traffic conditions, it should define which traffic should be allowed when no more service units are granted. However, if QoS or treatment information exists in the AVP, these actions should be executed, e.g., limiting the allowed traffic with certain QoS.

When multiple Filter-Rule AVPs exist, precedence should be determined as defined in [RFC5777].

The Filter-Id AVP is defined in [RFC7155]. 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.

If the Final-Unit-Action AVP is set to TERMINATE, or set to RESTRICT_ACCESS and the action required is allow only traffic that could be classified using an IPFilterRule, or set to REDIRECT of a type which is one of the types in the Redirect-Address-Type AVP, then the credit-control server SHOULD send the information in the Final-Unit-Indication AVP, in addition to or instead of the QoS-Final-Unit-Indication AVP. This is in order to preserve backward compatibility with credit-control clients that support only [RFC4006].

The QoS-Final-Unit-Indication AVP is defined as follows (per the grouped-avp-def of [RFC6733]):

```
QoS-Final-Unit-Indication ::= < AVP Header: TBD17 >
    { Final-Unit-Action }
    *[ Filter-Rule ]
    *[ Filter-Id ]
    [ Redirect-Server-Extension ]
    *[ AVP ]
```

9. Result Code AVP Values

This section defines new Result-Code AVP [RFC6733] 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
QoS-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
Subscription-Id-Extension	0+	0
Termination-Cause	0-1	0
User-Equipment-Info	0-1	0
User-Equipment-Info-Extension	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 [RFC6733].

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 Figure 9, and interworking flow in Figure 10. 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.

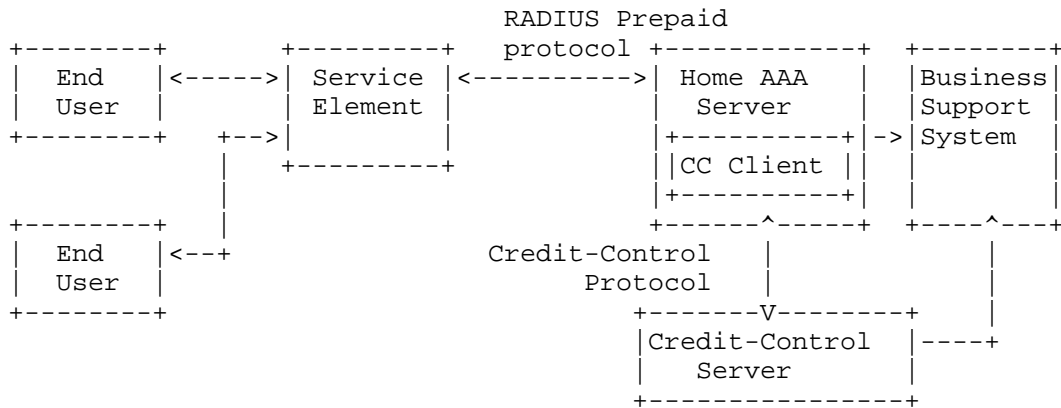


Figure 9: Credit-control architecture with service element containing translation agent, translating RADIUS prepaid to Diameter credit-control protocol

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.

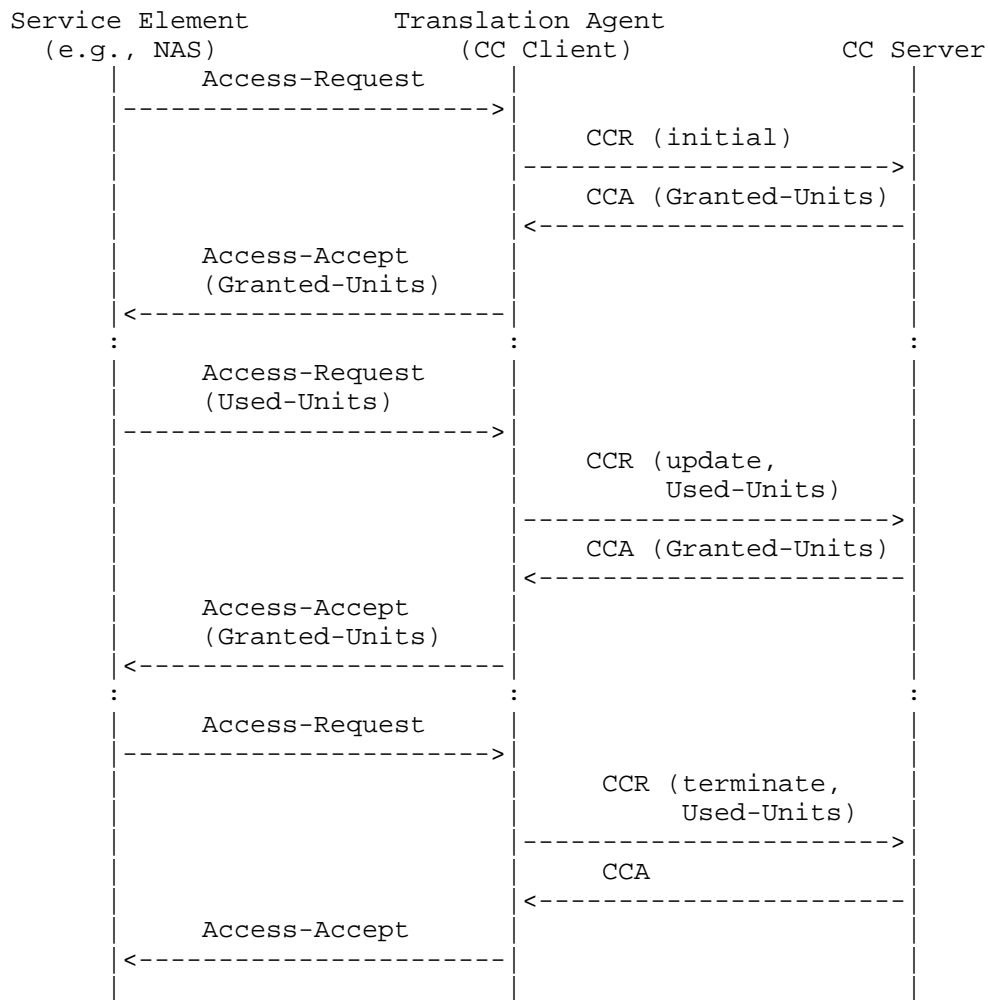


Figure 10: Message flow example with RADIUS prepaid - Diameter credit-control interworking

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 [RFC6733]. See Section 1.3 for more information.

12.2. Command Codes

This specification uses the value 272 from the Command code namespace defined in [RFC6733] 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 [RFC6733]. 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 [RFC6733]. 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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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

Security considerations regarding the Diameter protocol itself are discussed in [RFC6733]. Use of this application of Diameter MUST take into consideration the security issues and requirements of the base protocol.

This application includes a mechanism for application layer replay protection by means of the Session-Id from [RFC6733] 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/TCP, DTLS/SCTP or IPsec is sufficient. The details of TLS/TCP, DTLS/SCTP or IPsec related security considerations are discussed in the [RFC6733].

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 [RFC6733], 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

- [CE164] "Complement to ITU-T Recommendation E.164 (05/1997):"List of ITU-T Recommendation E.164 assigned country codes", June 2000.

- [CE212] "Complement to ITU-T Recommendation E.212 (11/1997):" List of mobile country or geographical area codes", February 1999.
- [E164] "Recommendation E.164/I.331 (05/97): The International Public Telecommunication Numbering Plan.", 1997.
- [E212] "Recommendation E.212 (11/98): The international identification plan for mobile terminals and mobile users.", 1998.
- [EUI64] IEEE, "'Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority'", March 1997, <<http://standards.ieee.org/regauth/oui/tutorials/EUI64.html>>.
- [ISO4217] "Codes for the representation of currencies and funds, International Standard ISO 4217", 2001.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [RFC1738] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, DOI 10.17487/RFC1738, December 1994, <<http://www.rfc-editor.org/info/rfc1738>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, DOI 10.17487/RFC2434, October 1998, <<http://www.rfc-editor.org/info/rfc2434>>.
- [RFC2486] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, DOI 10.17487/RFC2486, January 1999, <<http://www.rfc-editor.org/info/rfc2486>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.

- [RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, DOI 10.17487/RFC3539, June 2003, <<http://www.rfc-editor.org/info/rfc3539>>.
- [RFC3580] 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, DOI 10.17487/RFC3580, September 2003, <<http://www.rfc-editor.org/info/rfc3580>>.
- [RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, "Diameter Credit-Control Application", RFC 4006, DOI 10.17487/RFC4006, August 2005, <<http://www.rfc-editor.org/info/rfc4006>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, DOI 10.17487/RFC5777, February 2010, <<http://www.rfc-editor.org/info/rfc5777>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<http://www.rfc-editor.org/info/rfc5952>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.
- [RFC7155] Zorn, G., Ed., "Diameter Network Access Server Application", RFC 7155, DOI 10.17487/RFC7155, April 2014, <<http://www.rfc-editor.org/info/rfc7155>>.
- [TGPPCHARG] 3rd Generation Partnership Project, "Technical Specification Group Services and System Aspects, Service aspects; Charging and Billing, (release 13), 3GPP TS 22.115 v. 13.3.0", 2016-03.

[TGPPIMEI]

3rd Generation Partnership Project, "Technical Specification Group Core Network, Numbering, addressing and identification, (release 13), 3GPP TS 23.003 v. 13.5.0", 2016-04.

15.2. Informative References

- [DIAMEAP] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", Work in Progress.
- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, DOI 10.17487/RFC2866, June 2000, <<http://www.rfc-editor.org/info/rfc2866>>.
- [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, DOI 10.17487/RFC3725, April 2004, <<http://www.rfc-editor.org/info/rfc3725>>.
- [RFC4004] Calhoun, P., Johansson, T., Perkins, C., Hiller, T., Ed., and P. McCann, "Diameter Mobile IPv4 Application", RFC 4004, DOI 10.17487/RFC4004, August 2005, <<http://www.rfc-editor.org/info/rfc4004>>.

Appendix A. Acknowledgements

The original authors of RFC4006 are: Harri Hakala, Leena Mattila, Juha-Pekka Koskinen, Marco Stura, and John Loughney.

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 B. Credit-Control Sequences

B.1. Flow I

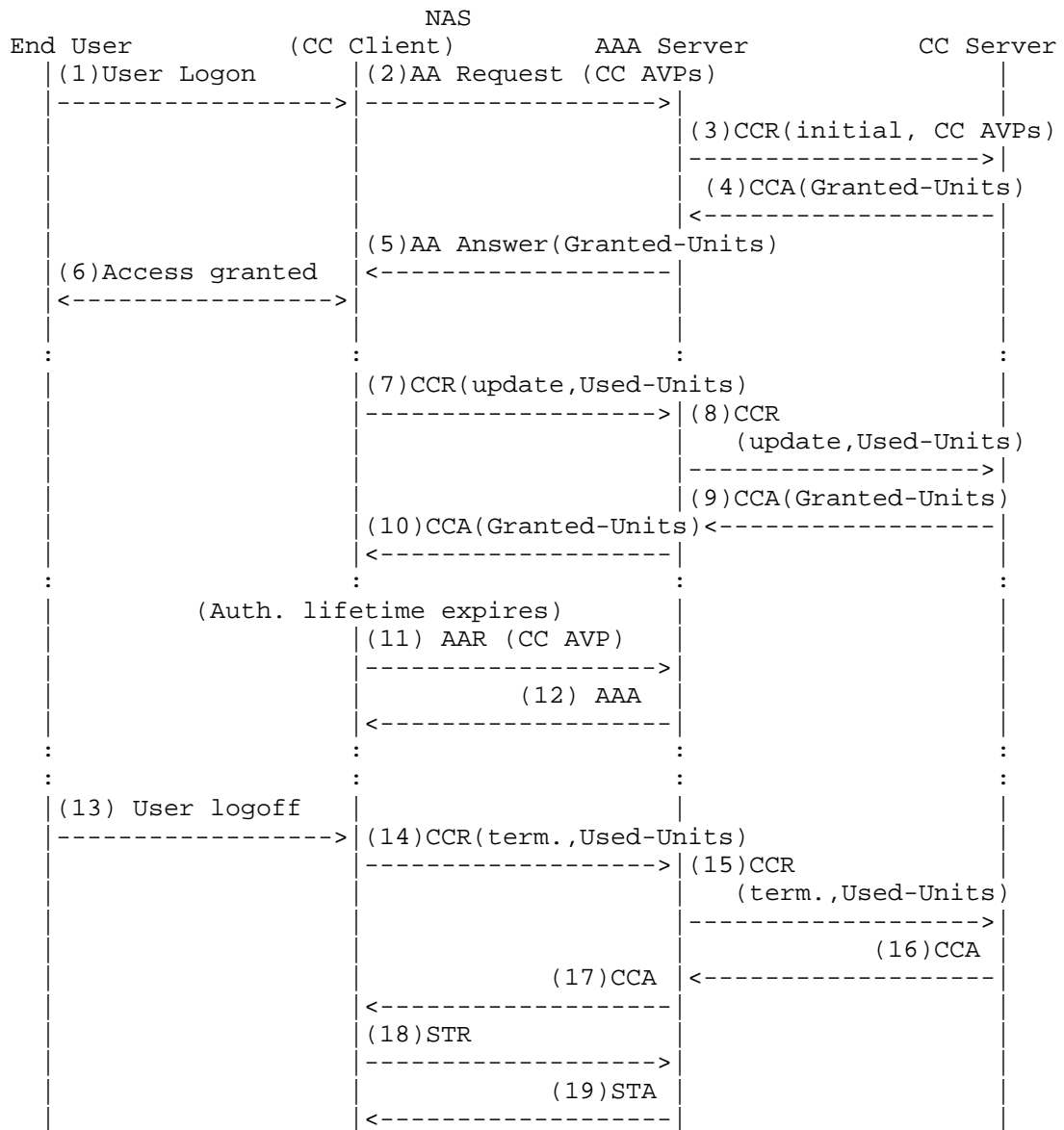


Figure 11: Flow I

A credit-control flow for Network Access Services prepaid is shown in Figure 11. The Diameter [RFC7155] 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 [RFC7155]. 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).

B.2. Flow II

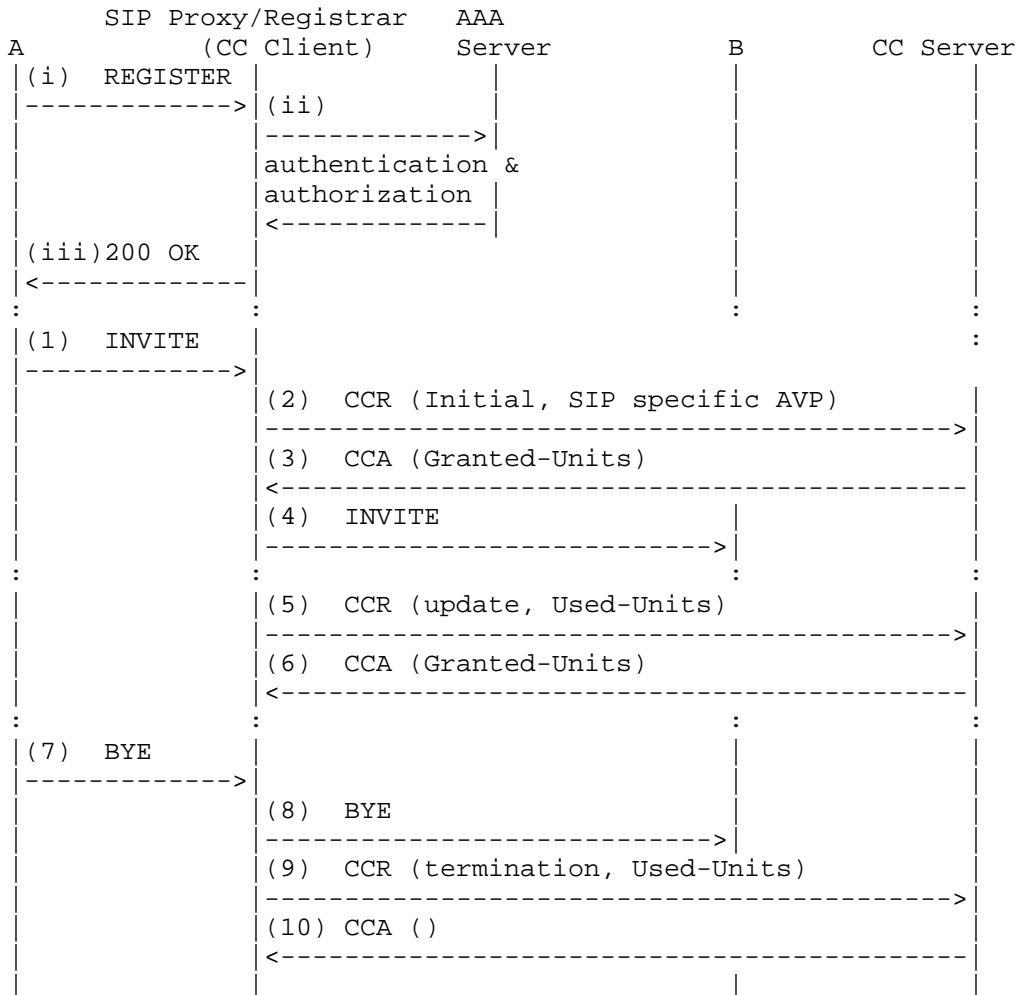


Figure 12: Flow II

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 [RFC3261]). 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).

B.3. Flow III

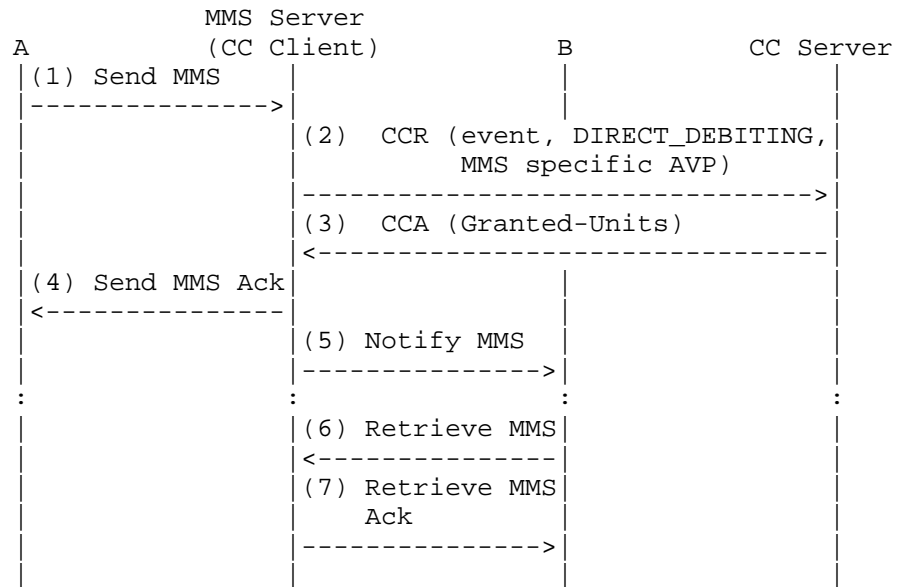


Figure 13: Flow III

A credit-control flow for Multimedia Messaging Services is shown in Figure 13. 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).

B.4. Flow IV

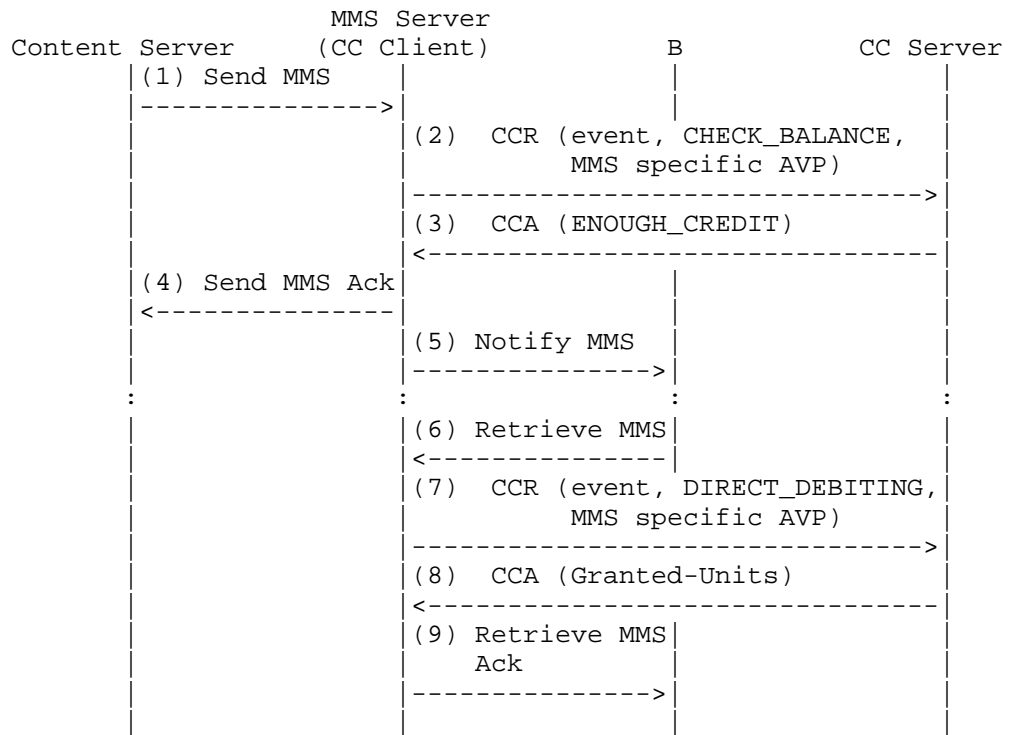


Figure 14: Flow IV

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 14. 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.

B.5. Flow V

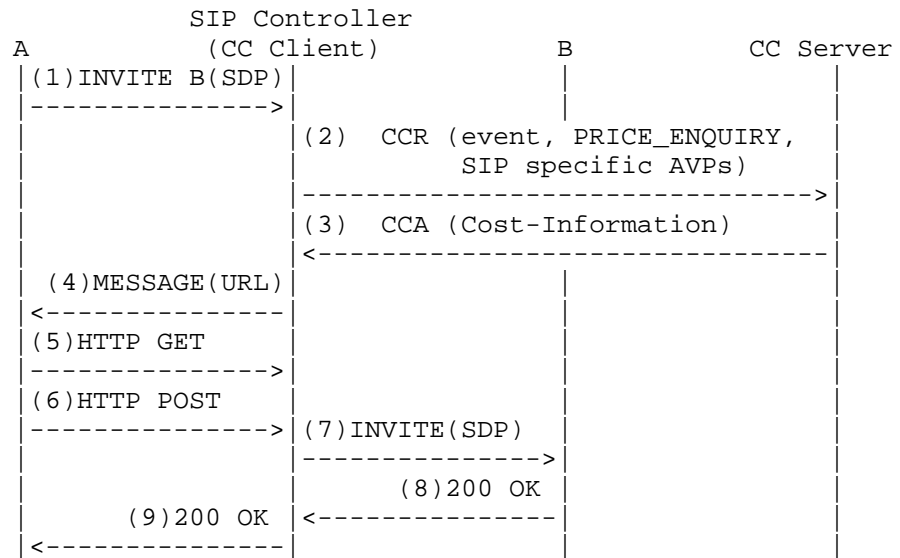


Figure 15: Flow V

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 15 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).

B.6. Flow VI

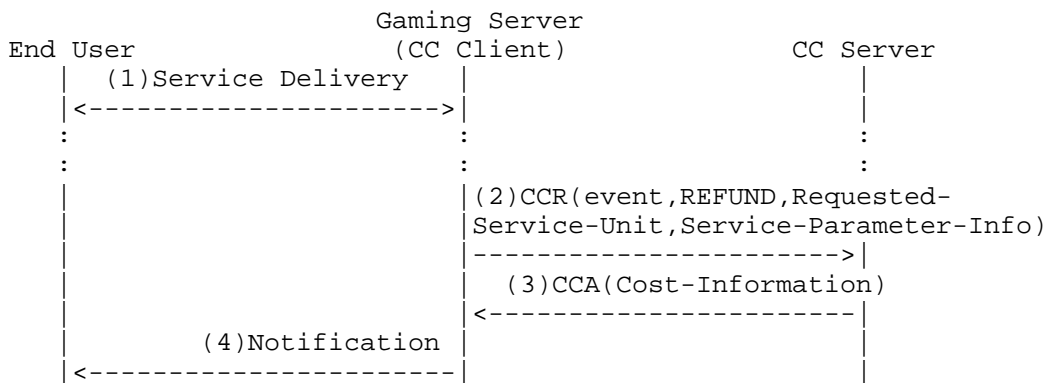


Figure 16: Flow VI

Figure 16 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 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).

B.7. Flow VII

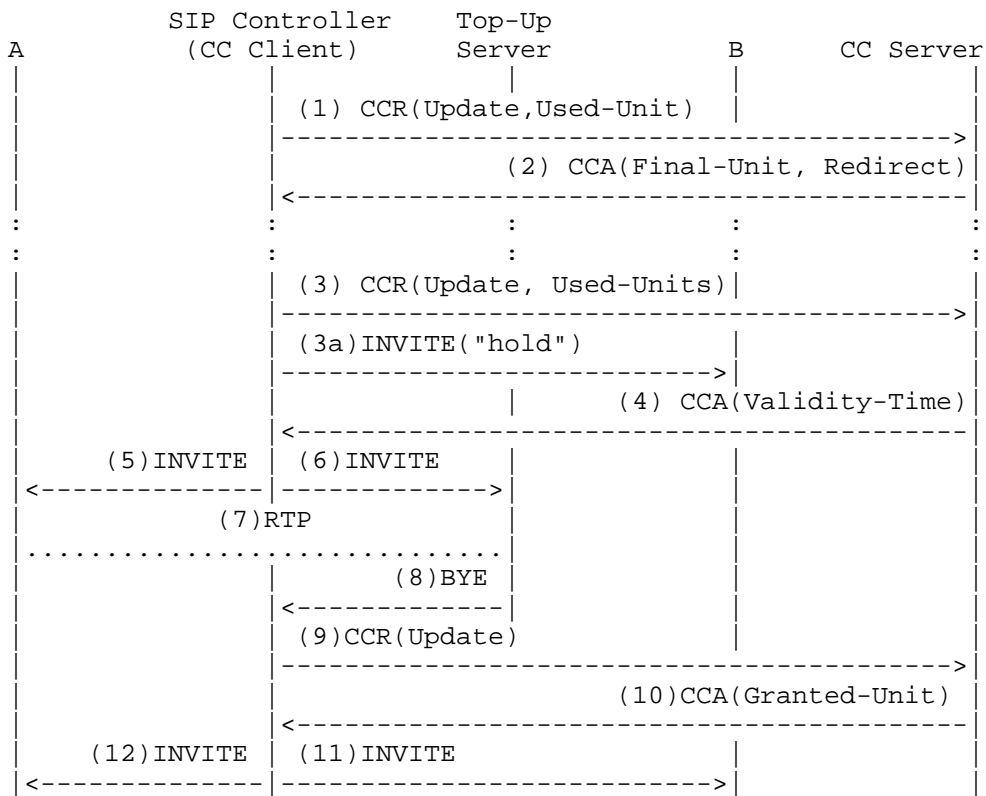


Figure 17: Flow VII

Figure 17 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).

B.8. Flow VIII

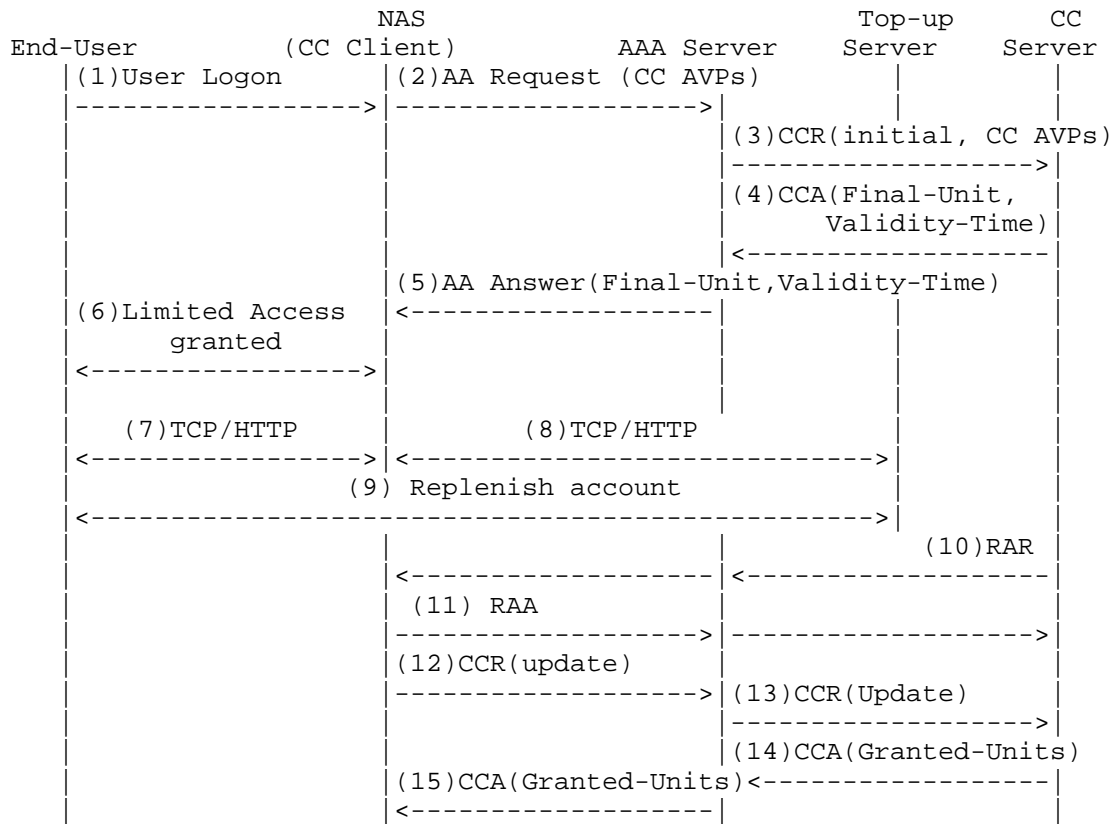


Figure 18: Flow VIII

Figure 18 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 [RFC7155] 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 [RFC7155]. 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

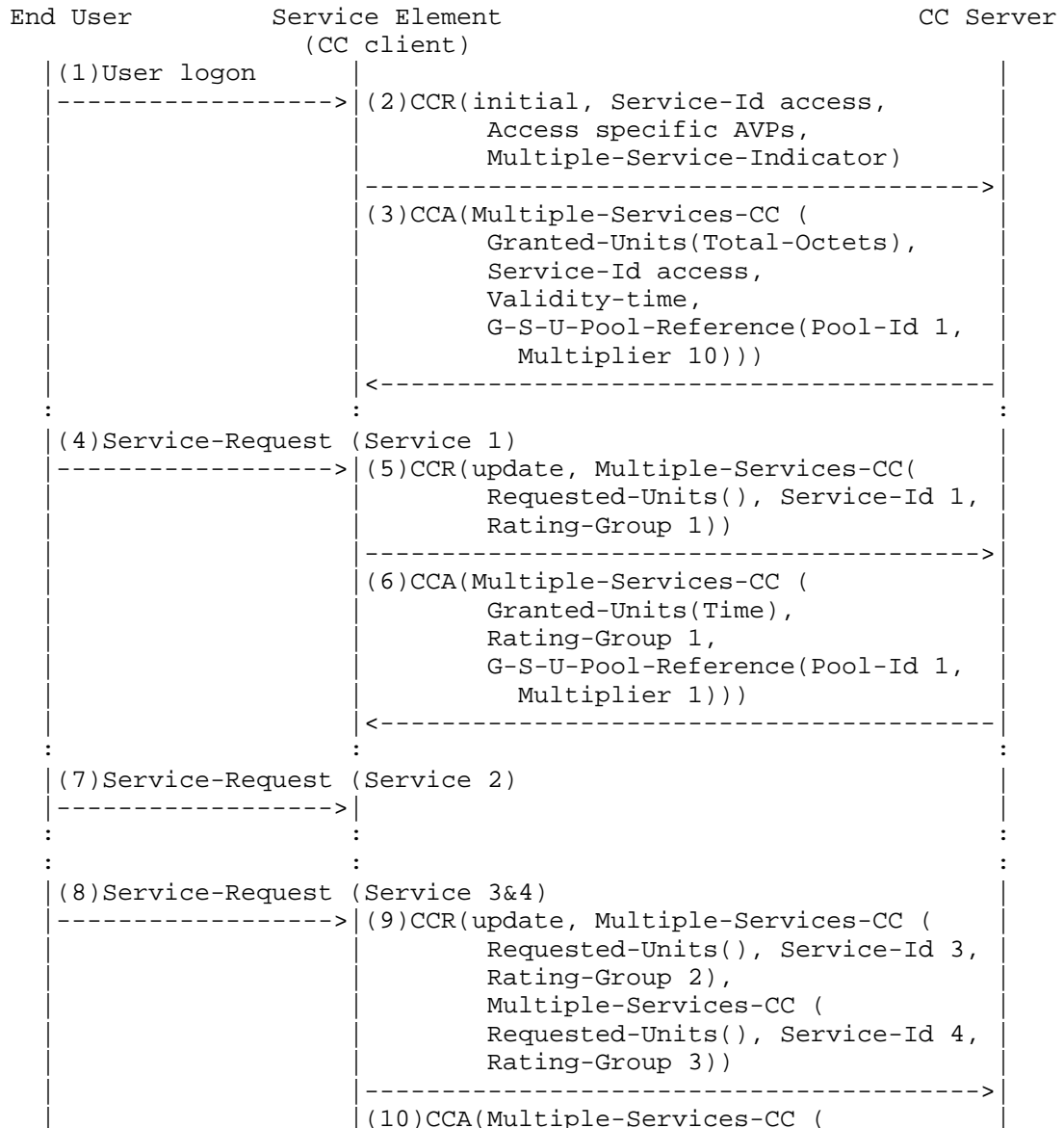
establish a credit-control session. (The home Diameter AAA server may forward service specific AVPs received from the NAS as input for the rating process.) The Diameter credit-control server checks the end user's account balance, determines that the account cannot cover the cost of the service, and initiates 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.

B.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.




```

----->| (16)CCR(term,
         |         Multiple-Services-CC (
         |             Used-Units(In-Octets,Out-Octets),
         |             Service-Id access),
         |         Multiple-Services-CC (
         |             Used-Units(Time),
         |             Service-Id 1, Rating-Group 1),
         |         Multiple-Services-CC (
         |             Used-Units(Time),
         |             Service-Id 2, Rating-Group 1))
         |----->
         | (17)CCA(term)
         |<-----

```

Figure 19: 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 to 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 to 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).

Appendix C. Changes relative to RFC4006

The following changes were made relative to RFC4006:

- Update references to obsolete RFC 3588 to refer to RFC 6733.
- Update references to obsolete RFC 4005 to refer to RFC 7155.
- Update references to current 3GPP documents.
- Update AVP per Errata ID 3329.
- Update reference to "IPsec or TLS" to be "TLS/TCP, DTLS/SCTP or IPsec".
- Clarify Filter-Rule AVP in Restrict Access Action.

Remove Encr column from AVP flag rules.

Clarify that RESTRICT_ACCESS action applies after consumption of final granted units (Section 5.6.3).

Clarify that values in Used-Service-Unit AVP may exceed Granted-Service-Unit AVP (Section 8.19).

Clarify that IPv6 representation in Redirect-Address-Type AVP conforms to RFC5952 (Section 8.38).

Describe immediate graceful service termination procedure (in Section 5.6).

Add extensible User-Equipment-Info-Extension AVP and included types (from Section 8.52 to Section 8.57).

Add extensible Subscription-Id-Extension AVP and included types (from Section 8.58 to Section 8.63).

Add extensible Redirect-Server-Extension AVP and included types (from Section 8.64 to Section 8.67).

Add extensible QoS-Final-Unit-Indication AVP (in Section 8.68).

Updated Security Section to include language consistent with structures of latest base protocol specification.

Authors' Addresses

Lyle Bertz (editor)
Sprint
6220 Sprint Parkway
Overland Park, KS 66251
United States

Email: lyleb551144@gmail.com

David Dolson (editor)
Sandvine
408 Albert Street
Waterloo, ON N2L 3V3
Canada

Phone: +1 519 880 2400
Email: ddolson@sandvine.com

Yuval Lifshitz (editor)
Sandvine
408 Albert Street
Waterloo, ON N2L 3V3
Canada

Phone: +1 519 880 2400
Email: ylifshitz@sandvine.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 11, 2017

L. Bertz, Ed.
Sprint
D. Dolson, Ed.
Y. Lifshitz, Ed.
Sandvine
May 10, 2017

Diameter Credit-Control Application
draft-ietf-dime-rfc4006bis-03

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.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 11, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	5
1.1.	Requirements Language	6
1.2.	Terminology	6
1.3.	Advertising Application Support	8
2.	Architecture Models	8
3.	Credit-Control Messages	10
3.1.	Credit-Control-Request (CCR) Command	10
3.2.	Credit-Control-Answer (CCA) Command	11
4.	Credit-Control Application Overview	12
4.1.	Service-Specific Rating Input and Interoperability	14
4.1.1.	Specifying Rating Input AVPs	14
4.1.2.	Service-Specific Documentation	15
4.1.3.	Handling of Unsupported/Incorrect Rating Input	16
4.1.4.	RADIUS Vendor-Specific Rating Attributes	16
5.	Session Based Credit-Control	16
5.1.	General Principles	16
5.1.1.	Basic Tariff-Time Change Support	17
5.1.2.	Credit-Control for Multiple Services within a (sub-)Session	18
5.2.	First Interrogation	22
5.2.1.	First Interrogation after Authorization and Authentication	24
5.2.2.	Authorization Messages for First Interrogation	25
5.3.	Intermediate Interrogation	28
5.4.	Final Interrogation	30
5.5.	Server-Initiated Credit Re-Authorization	31
5.6.	Graceful Service Termination	33
5.6.1.	Terminate Action	36
5.6.2.	Redirect Action	37
5.6.3.	Restrict Access Action	39
5.6.4.	Usage of the Server-Initiated Credit Re-Authorization	40

5.7.	Failure Procedures	40
6.	One Time Event	43
6.1.	Service Price Enquiry	44
6.2.	Balance Check	45
6.3.	Direct Debiting	45
6.4.	Refund	46
6.5.	Failure Procedure	47
7.	Credit-Control Application State Machine	49
8.	Credit-Control AVPs	57
8.1.	CC-Correlation-Id AVP	60
8.2.	CC-Request-Number AVP	60
8.3.	CC-Request-Type AVP	60
8.4.	CC-Session-Failover AVP	61
8.5.	CC-Sub-Session-Id AVP	61
8.6.	Check-Balance-Result AVP	62
8.7.	Cost-Information AVP	62
8.8.	Unit-Value AVP	63
8.9.	Exponent AVP	63
8.10.	Value-Digits AVP	63
8.11.	Currency-Code AVP	63
8.12.	Cost-Unit AVP	64
8.13.	Credit-Control AVP	64
8.14.	Credit-Control-Failure-Handling AVP	64
8.15.	Direct-Debiting-Failure-Handling AVP	65
8.16.	Multiple-Services-Credit-Control AVP	66
8.17.	Granted-Service-Unit AVP	67
8.18.	Requested-Service-Unit AVP	68
8.19.	Used-Service-Unit AVP	68
8.20.	Tariff-Time-Change AVP	69
8.21.	CC-Time AVP	69
8.22.	CC-Money AVP	69
8.23.	CC-Total-Octets AVP	70
8.24.	CC-Input-Octets AVP	70
8.25.	CC-Output-Octets AVP	70
8.26.	CC-Service-Specific-Units AVP	70
8.27.	Tariff-Change-Usage AVP	70
8.28.	Service-Identifier AVP	71
8.29.	Rating-Group AVP	71
8.30.	G-S-U-Pool-Reference AVP	71
8.31.	G-S-U-Pool-Identifier AVP	72
8.32.	CC-Unit-Type AVP	72
8.33.	Validity-Time AVP	72
8.34.	Final-Unit-Indication AVP	73
8.35.	Final-Unit-Action AVP	74
8.36.	Restriction-Filter-Rule AVP	75
8.37.	Redirect-Server AVP	75
8.38.	Redirect-Address-Type AVP	76
8.39.	Redirect-Server-Address AVP	76

8.40.	Multiple-Services-Indicator AVP	76
8.41.	Requested-Action AVP	77
8.42.	Service-Context-Id AVP	78
8.43.	Service-Parameter-Info AVP	78
8.44.	Service-Parameter-Type AVP	79
8.45.	Service-Parameter-Value AVP	79
8.46.	Subscription-Id AVP	79
8.47.	Subscription-Id-Type AVP	80
8.48.	Subscription-Id-Data AVP	80
8.49.	User-Equipment-Info AVP	81
8.50.	User-Equipment-Info-Type AVP	81
8.51.	User-Equipment-Info-Value AVP	82
8.52.	User-Equipment-Info-Extension AVP	82
8.53.	User-Equipment-Info-IMEISV AVP	82
8.54.	User-Equipment-Info-MAC AVP	82
8.55.	User-Equipment-Info-EUI64 AVP	82
8.56.	User-Equipment-Info-ModifiedEUI64 AVP	83
8.57.	User-Equipment-Info-IMEI AVP	83
8.58.	Subscription-Id-Extension AVP	83
8.59.	Subscription-Id-E164 AVP	84
8.60.	Subscription-Id-IMSI AVP	84
8.61.	Subscription-Id-SIP-URI AVP	84
8.62.	Subscription-Id-NAI AVP	84
8.63.	Subscription-Id-Private AVP	84
8.64.	Redirect-Server-Extension AVP	84
8.65.	Redirect-Address-IPAddress AVP	85
8.66.	Redirect-Address-URL AVP	85
8.67.	Redirect-Address-SIP-URI AVP	85
8.68.	QoS-Final-Unit-Indication AVP	86
9.	Result Code AVP Values	87
9.1.	Transient Failures	87
9.2.	Permanent Failures	88
10.	AVP Occurrence Table	88
10.1.	Credit-Control AVP Table	89
10.2.	Re-Auth-Request/Answer AVP Table	90
11.	RADIUS/Diameter Credit-Control Interworking Model	90
12.	IANA Considerations	93
12.1.	Application Identifier	94
12.2.	Command Codes	94
12.3.	AVP Codes	94
12.4.	Result-Code AVP Values	94
12.5.	CC-Request-Type AVP	94
12.6.	CC-Session-Failover AVP	94
12.7.	CC-Unit-Type AVP	94
12.8.	Check-Balance-Result AVP	95
12.9.	Credit-Control AVP	95
12.10.	Credit-Control-Failure-Handling AVP	95
12.11.	Direct-Debiting-Failure-Handling AVP	95

12.12. Final-Unit-Action AVP	95
12.13. Multiple-Services-Indicator AVP	95
12.14. Redirect-Address-Type AVP	95
12.15. Requested-Action AVP	96
12.16. Subscription-Id-Type AVP	96
12.17. Tariff-Change-Usage AVP	96
12.18. User-Equipment-Info-Type AVP	96
13. Credit-Control Application Related Parameters	96
14. Security Considerations	97
14.1. Direct Connection with Redirects	98
15. References	98
15.1. Normative References	98
15.2. Informative References	101
Appendix A. Acknowledgements	101
Appendix B. Credit-Control Sequences	101
B.1. Flow I	101
B.2. Flow II	104
B.3. Flow III	106
B.4. Flow IV	106
B.5. Flow V	108
B.6. Flow VI	109
B.7. Flow VII	110
B.8. Flow VIII	112
B.9. Flow IX	114
Appendix C. Changes relative to RFC4006	119
Authors' Addresses	120

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 [TGPPCHARG] 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 [RFC2119].

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., [RFC7155] and [RFC4004].

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., [RFC7155] and [RFC4004].

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 [RFC6733].

2. Architecture Models

The current accounting models specified in the Radius Accounting [RFC2866] and Diameter base [RFC6733] 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, [RFC7155] and [RFC4004], 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.

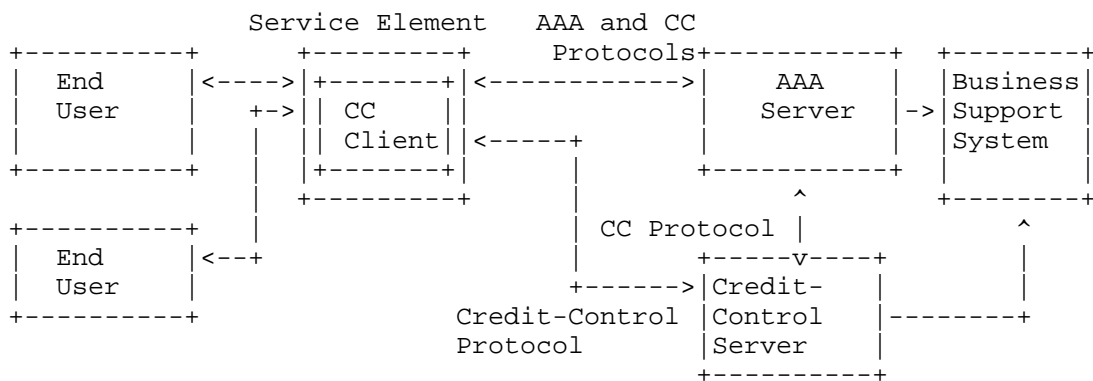


Figure 1: Typical credit-control architecture

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 [RFC6733], 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

Table 1: Credit-Control Commands

Diameter Base [RFC6733] defines in the section 3.2 the Command Code format 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 ]
    *[ Subscription-Id-Extension ]
    [ 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 ]
    [ User-Equipment-Info-Extension ]
    *[ 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 ]
    [ QoS-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., [RFC7155] for network access services). Re-use of existing AVPs is strongly recommended in [RFC6733].

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 [RFC6733], section 1.3.

1b. New AVPs can be defined if the existing AVPs do not provide sufficient rating information. In this case, the procedures defined in [RFC6733] 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 [RFC6733].

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.

4.1.2. 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, or 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 [RFC6733].

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 [RFC7155] 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 Figure 3 and Figure 4.

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 [RFC6733], 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.

5.1.1.1. 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.

5.1.1.2. 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 or QoS-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.

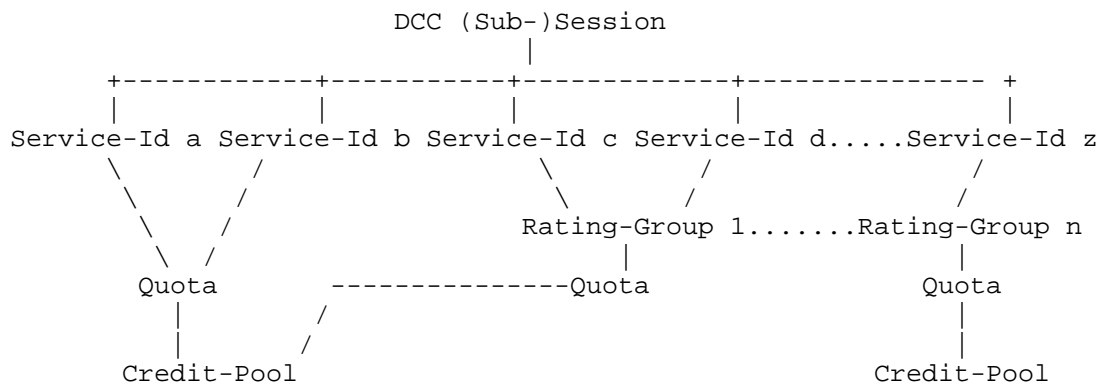


Figure 2: Multiple-Service (sub)-Session Example

If independent credit-control of multiple services is used, the Validity-Time AVP and Final-Unit-Indication AVP or QoS-Final-Unit-Indication AVP 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 AVP on the command level indicates a value other than SUCCESS, then

the Result-Code AVP 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:

$$S = Q_1 * M_1 + Q_2 * M_2 + \dots + Q_n * M_n$$

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 or the Subscription-Id-Extension 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 or User-Equipment-Info-Extension 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 or the QoS-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-control 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.

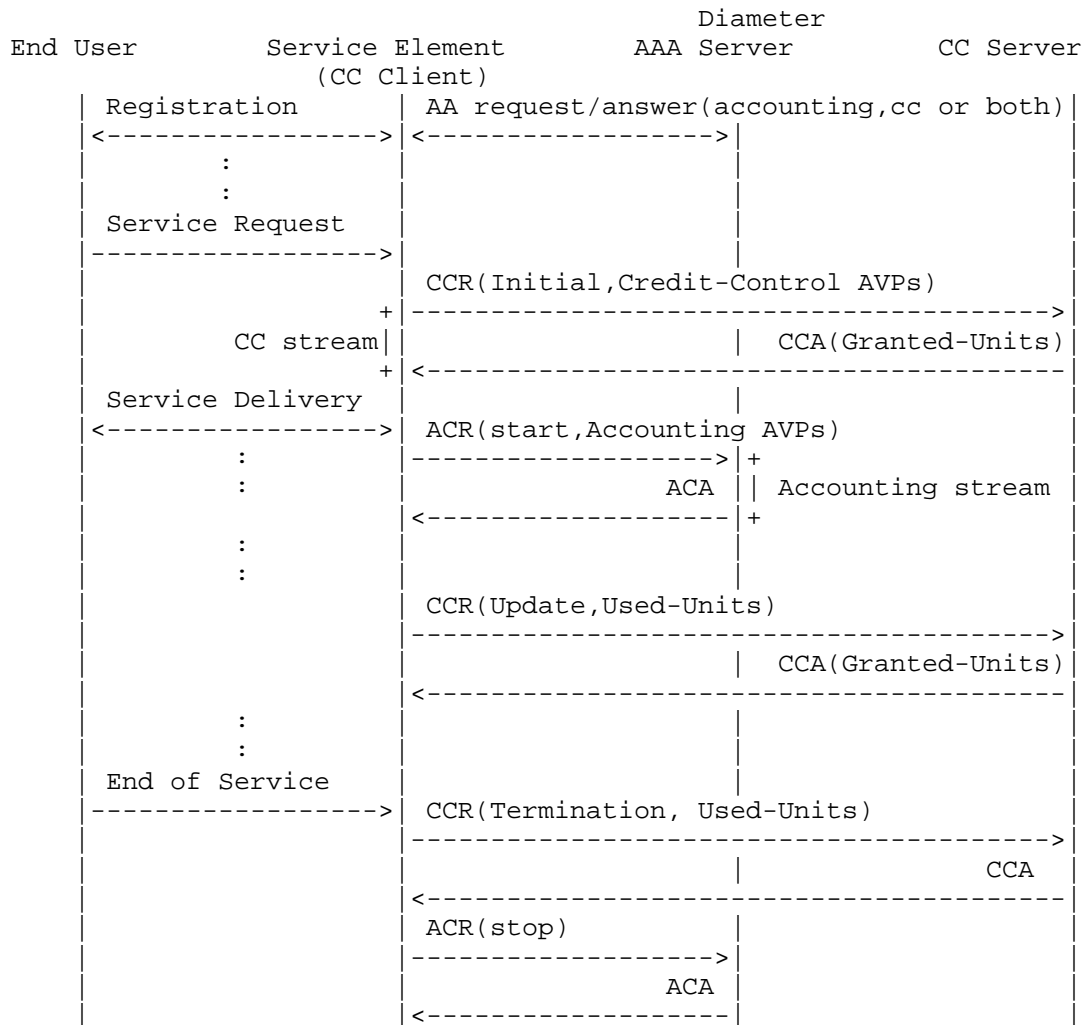


Figure 3: Protocol example with first interrogation after user's authorization/authentication

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., [RFC7155], [RFC4004]). 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., [RFC7155], [RFC4004]). 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.

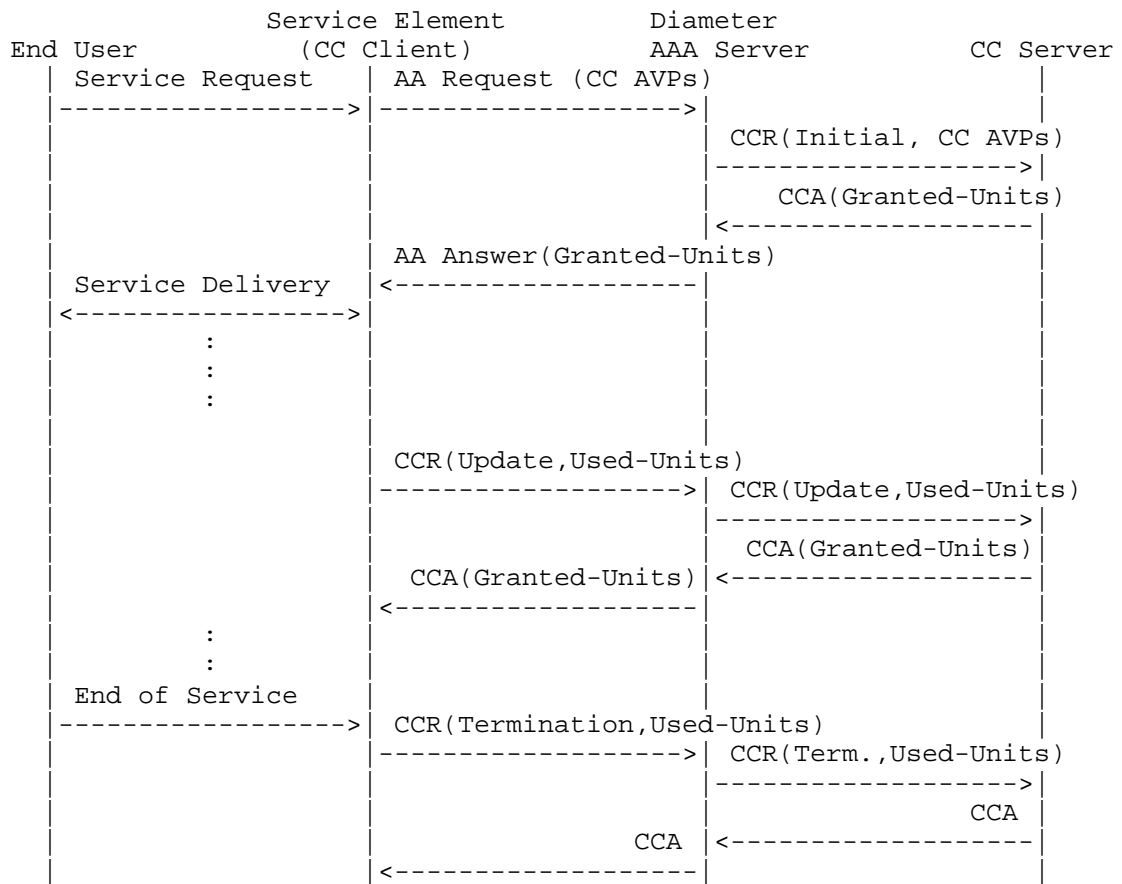


Figure 4: Protocol example with use of the authorization messages for the first interrogation

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 or Subscription-Id-Extension 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 or the QoS-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 [RFC6733]. Figure 5 illustrates the case when the final-unit

indication causes user logoff upon consumption of the final granted units and the generation of STR.

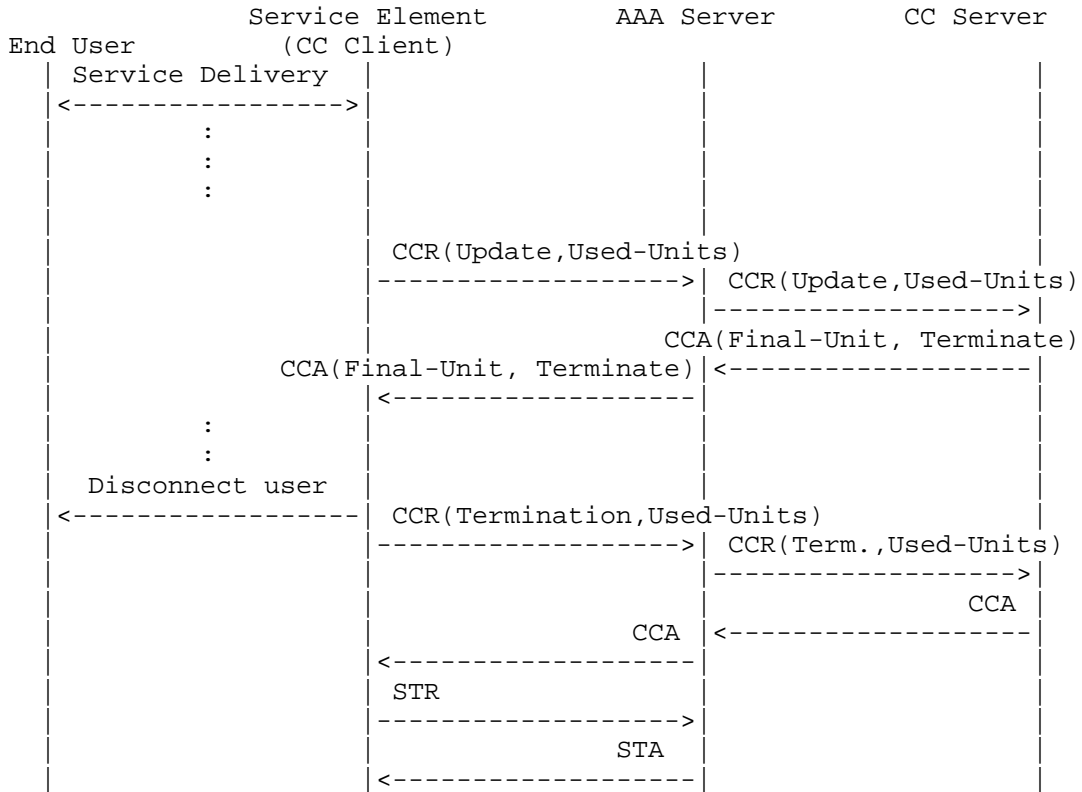


Figure 5: User disconnected due to exhausted account

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 [RFC6733]. 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 AVP or QoS-Final-Unit-Indication AVP 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 Appendix B.7.

The credit-control server MAY initiate the graceful service termination by including the Final-Unit-Indication AVP or the QoS-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 or the QoS-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.

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

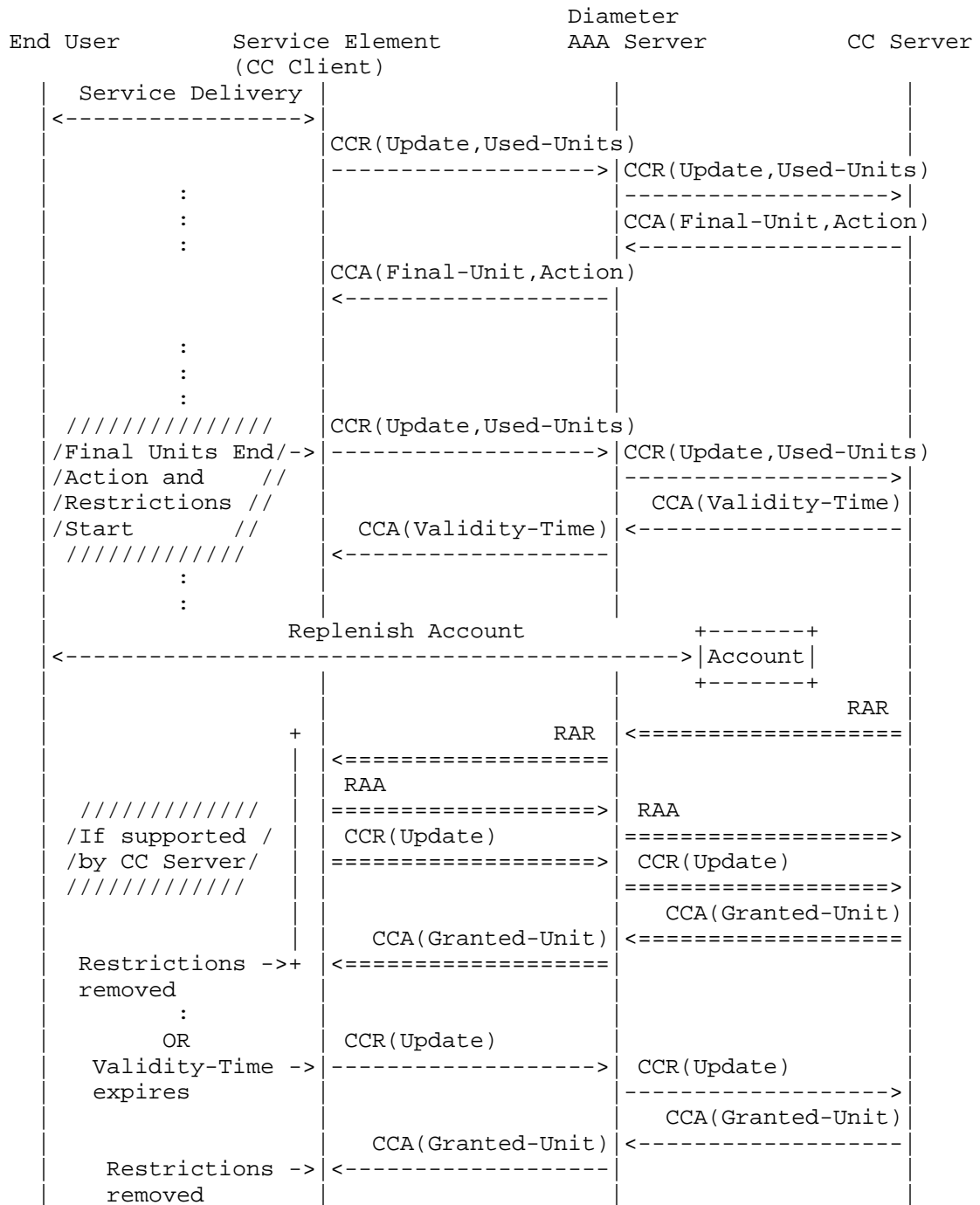


Figure 6: Optional graceful service termination procedure

In addition, the credit-control server MAY reply with Final-Unit-Indication AVP or QoS-Final-Unit-Indication AVP holding a G-S-U AVP with a zero grant, indicating that the service SHOULD be terminated immediately, and no further reporting is required. A following figure illustrates a graceful service termination procedure that applies immediately after receiving a zero grant.

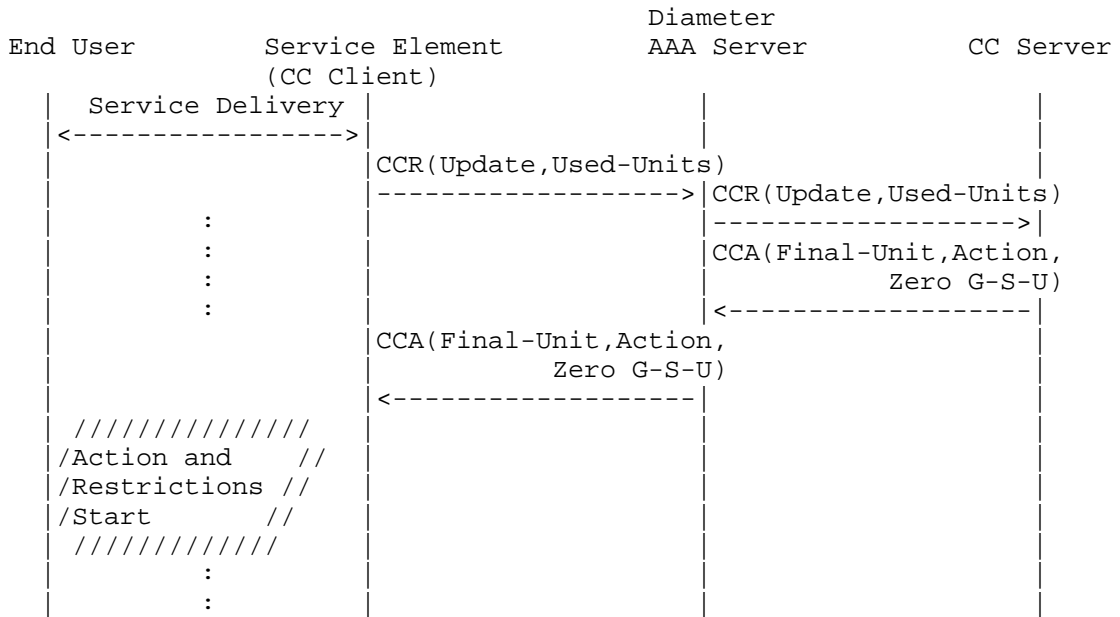


Figure 7: Optional immediate graceful service termination procedure

5.6.1. Terminate Action

The Final-Unit-Indication AVP or the QoS-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 or the QoS-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 or the QoS-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 or Redirect-Server-Extension AVP as follows.

The credit-control server sends the Redirect-Server AVP or Redirect-Server-Extension 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 or Redirect-Server-Extension 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 or Redirect-Server-Extension AVP, the credit-control server MAY include one or more Restriction-Filter-Rule AVPs, one or more 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 Restriction-Filter-Rule AVPs, Filter-Rule AVPs or Filter-Id AVPs. If enforcement actions other than allowing the packets (e.g., QoS), are indicated in the Filter-Rule AVPs or Filter-Id AVPs, they SHOULD be performed as well. In addition, if possible, to redirecting the user to the destination specified in the Redirect-Server AVP or Redirect-Server-Extension 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 B.8 (Figure 18).

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 or the QoS-Final-Unit-Indication AVP and Validity-Time AVPs but no Granted-Service-Unit AVP. It immediately starts the graceful service termination without sending any message to the server. An example of this case is illustrated in Appendix B.

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, upon consumption of the final granted units, 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 Final-Unit-Indication group AVP of the Credit-Control-Answer message.

A QoS-Final-Unit-Indication AVP with the Final-Unit-Action RESTRICT_ACCESS indicates to the device supporting this action that, upon consumption of the final granted units, the actions specified in Filter-Rule AVP(s) MUST restrict the traffic according to the classifiers in the Filter-Rule AVP(s). If Filter-Id AVP(s) are provided in the Credit-Control-Answer message, the credit control client MUST restrict the traffic according to the IP packet filters identified by the Filter-Id AVP(s). The credit-control server SHOULD include either the Filter-Rule AVP or the Filter-Id AVP in the QoS-Final-Unit-Indication group AVP of the Credit-Control-Answer message.

If both Final-Unit-Indication AVP and QoS-Final-Unit-Indication AVP exist in the Credit-Control-Answer message, a credit control client which supports the QoS-Final-Unit-Indication AVP SHOULD follow the directives included in the QoS-Final-Unit-Indication AVP and SHOULD ignore the Final-Unit-Indication 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. 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-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 [RFC6733]. 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 an 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 [RFC6733]. 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 [RFC3539] 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 [RFC3539]. The recommended default initial value for Tw (Twinit) is 30 seconds. Twinit may be set as low as 6 seconds; however, according to [RFC3539], 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 failback 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 8.

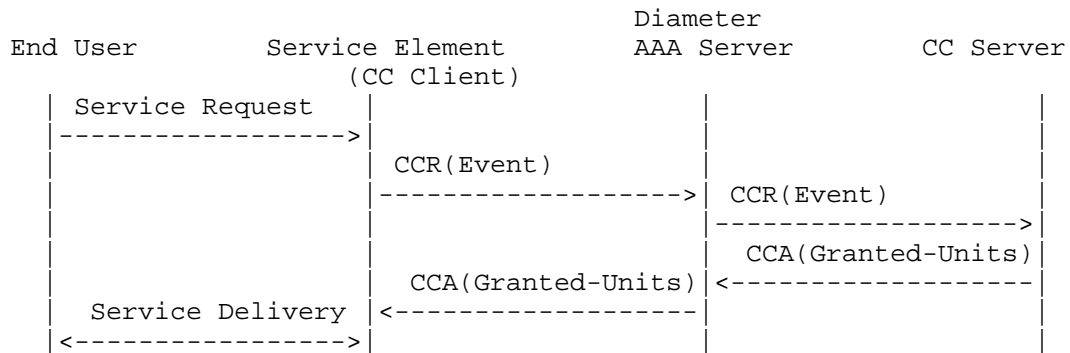


Figure 8: One time event

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 services 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 or Subscription-Id-Extension 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 or Subscription-Id-Extension 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 or Subscription-Id-Extension 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 [RFC6733]. 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 [RFC6733], 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 [RFC6733], 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 [RFC6733], 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 [RFC6733].

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 [RFC6733].

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.

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

Table 2: CLIENT, SESSION BASED for the first interrogation with AA request

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

Table 3: CLIENT, SESSION BASED for the first interrogation with CCR

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

Table 4: CLIENT, SESSION BASED for intermediate and final interrogations

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	Indicate service	Idle

	received, or Tx expired; requested action CHECK_BALANCE or PRICE_ENQUIRY	error	
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, 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

Table 5: CLIENT, 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

Table 6: SERVER, SESSION AND EVENT BASED

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 [RFC7155] and [RFC4004], 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 [RFC6733], 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, and possible flag values. The AVP Flag rules are explained in the Diameter base [RFC6733], section 4.1.

Attribute Name	AVP Code	Section Defined	Data Type	AVP Flag rules		
				MUST	MAY	MUST NOT
CC-Correlation-Id	411	8.1	OctetString		M	V
CC-Input-Octets	412	8.24	Unsigned64	M		V
CC-Money	413	8.22	Grouped	M		V
CC-Output-Octets	414	8.25	Unsigned64	M		V
CC-Request-Number	415	8.2	Unsigned32	M		V
CC-Request-Type	416	8.3	Enumerated	M		V
CC-Service-Specific-Units	417	8.26	Unsigned64	M		V
CC-Session-Failover	418	8.4	Enumerated	M		V
CC-Sub-Session-Id	419	8.5	Unsigned64	M		V
CC-Time	420	8.21	Unsigned32	M		V
CC-Total-Octets	421	8.23	Unsigned64	M		V
CC-Unit-Type	454	8.32	Enumerated	M		V
Check-Balance-Result	422	8.6	Enumerated	M		V
Cost-Information	423	8.7	Grouped	M		V
Cost-Unit	424	8.12	UTF8String	M		V
Credit-Control	426	8.13	Enumerated	M		V
Credit-Control-Failure-Handling	427	8.14	Enumerated	M		V

Currency-Code	425	8.11	Unsigned32	M		V
Direct-Debiting- Failure-Handling	428	8.15	Enumerated	M		V
Exponent	429	8.9	Integer32	M		V
Final-Unit-Action	449	8.35	Enumerated	M		V
Final-Unit- Indication	430	8.34	Grouped	M		V
QoS-Final-Unit- Indication	TBD17	8.68	Grouped		M	V
Granted-Service- Unit	431	8.17	Grouped	M		V
G-S-U-Pool- Identifier	453	8.31	Unsigned32	M		V
G-S-U-Pool- Reference	457	8.30	Grouped	M		V
Multiple-Services -Credit-Control	456	8.16	Grouped	M		V
Multiple-Services -Indicator	455	8.40	Enumerated	M		V
Rating-Group	432	8.29	Unsigned32	M		V
Redirect-Address -Type	433	8.38	Enumerated	M		V
Redirect-Server	434	8.37	Grouped	M		V
Redirect-Server -Address	435	8.39	UTF8String	M		V
Redirect-Server -Extension	TBD13	8.64	Grouped		M	V
Redirect-Address -IPAddress	TBD14	8.65	Address		M	V
Redirect-Address -URL	TBD15	8.66	UTF8String		M	V
Redirect-Address -SIP-URI	TBD16	8.67	UTF8String		M	V
Requested-Action	436	8.41	Enumerated	M		V
Requested-Service -Unit	437	8.18	Grouped	M		V
Restriction -Filter-Rule	438	8.36	IPFilterRule	M		V
Service-Context -Id	461	8.42	UTF8String	M		V
Service- Identifier	439	8.28	Unsigned32	M		V
Service-Parameter -Info	440	8.43	Grouped		M	V
Service- Parameter-Type	441	8.44	Unsigned32		M	V
Service- Parameter-Value	442	8.45	OctetString		M	V

Subscription-Id	443	8.46	Grouped	M		V
Subscription-Id	444	8.48	UTF8String	M		V
-Data						
Subscription-Id	450	8.47	Enumerated	M		V
-Type						
Subscription-Id	TBD7	8.58	Grouped		M	V
-Extension						
Subscription-Id	TBD8	8.59	UTF8String		M	V
-E164						
Subscription-Id	TBD9	8.60	UTF8String		M	V
-IMSI						
Subscription-Id	TBD10	8.61	UTF8String		M	V
-SIP-URI						
Subscription-Id	TBD11	8.62	UTF8String		M	V
-NAI						
Subscription-Id	TBD12	8.63	UTF8String		M	V
-Private						
Tariff-Change	452	8.27	Enumerated	M		V
-Usage						
Tariff-Time	451	8.20	Time	M		V
-Change						
Unit-Value	445	8.8	Grouped	M		V
Used-Service-Unit	446	8.19	Grouped	M		V
User-Equipment	458	8.49	Grouped		M	V
-Info						
User-Equipment	459	8.50	Enumerated		M	V
-Info-Type						
User-Equipment	460	8.51	OctetString		M	V
-Info-Value						
User-Equipment	TBD1	8.52	Grouped		M	V
-Info-Extension						
User-Equipment	TBD2	8.53	OctetString		M	V
-Info-IMEISV						
User-Equipment	TBD3	8.54	OctetString		M	V
-Info-MAC						
User-Equipment	TBD4	8.55	OctetString		M	V
-Info-EUI64						
User-Equipment	TBD5	8.56	OctetString		M	V
-Info-ModifiedEUI64						
User-Equipment	TBD6	8.57	OctetString		M	V
-Info-IMEI						
Value-Digits	447	8.10	Integer64	M		V
Validity-Time	448	8.33	Unsigned32	M		V

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** 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 an 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 [RFC6733]):

```
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., Unit-Value = Value-Digits AVP * 10^{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 [RFC6733]):

```
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 [RFC6733] 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 or QoS-Final-Unit-Indication AVPs MAY be present in an answer command as defined in Section 5.1.2 and Section 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 [RFC6733]):

```
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 ]
    [ QoS-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 [RFC6733]):

```
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 [RFC6733]):

```
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. Note: The values reported in a Used-Service-Unit AVP does not necessarily have a relation to the grant provided in a Granted-Service-Unit AVP, e.g., the value in this AVP may exceed the value in the grant.

The Used-Service-Unit AVP is defined as follows (per the grouped-avp-def of [RFC6733]):

```
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 [RFC6733]):

```
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 B.9.

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 B.9.

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 [RFC6733]):

```
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, the Final-Unit-Indication group MUST NOT contain any other AVPs.

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 [RFC7155]. 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.

If the Final-Unit-Action AVP is set to REDIRECT and the type of server is not one of the enumerations in the Redirect-Address-Type AVP, then the QoS-Final-Unit-Indication AVP SHOULD be used together with the Redirect-Server-Extension AVP instead of the Final-Unit-Indication AVP.

If the Final-Unit-Action AVP is set to RESTRICT_ACCESS or REDIRECT and the classification of the restricted traffic cannot be expressed using IPFilterRule, or different actions (e.g., QoS) than just allowing QoS needs to be enforced traffic, then the QoS-Final-Unit-Indication AVP SHOULD be used instead of the Final-Unit-Indication AVP. However, if the credit control server wants to preserve backward compatibility with credit-control clients that support only [RFC4006], the Final-Unit-Indication AVP SHOULD be used together with the Filter-Id AVP.

The Final-Unit-Indication AVP is defined as follows (per the grouped-avp-def of [RFC6733]):

```
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 or one of the AVPs included in the Redirect-Server-Extension 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 filter AVPs contained in the applied grouped AVP: according to IP packet filters defined in the Restriction-Filter-Rule AVP, according to the packet classifier filters defined in Filter-Rule AVP, or according to the packet filters identified by the Filter-Id AVP. All the packets not matching any 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 [RFC6733]):

```
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 [RFC0791].

IPv6 Address 1

The address type is in the form of IPv6 address, as defined in [RFC4291]. The address MUST conform to the text representation of the address according to [RFC5952].

URL 2

The address type is in the form of Uniform Resource Locator, as defined in [RFC1738].

SIP URI 3

The address type is in the form of SIP Uniform Resource Identifier, as defined in [RFC3261].

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 [RFC6733]):

```
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 [RFC6733]):

```
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 [RFC6733].

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 [RFC3261].

END_USER_NAI 3

The identifier is in the form of a Network Access Identifier, as defined in [RFC7542].

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 [RFC6733]):

```
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 [TGPPIMEI].

MAC 1

The 48-bit MAC address is formatted as described in [RFC3580].

EUI64 2

The 64-bit identifier used to identify the 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 [RFC4291] 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.

8.52. User-Equipment-Info-Extension AVP

The User-Equipment-Info-Extension AVP (AVP Code TBD1) 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. If the type of the equipment is one of the enumerated types of User-Equipment-Info-Type AVP, then the credit-control client SHOULD send the information in the User-Equipment-Info AVP, in addition to or instead of the User-Equipment-Info-Extension AVP. This is in order to preserve backward compatibility with credit-control servers that support only RFC4006. Exactly one AVP MUST be included inside the User-Equipment-Info-Extension AVP.

It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
User-Equipment-Info-Extension ::= < AVP Header: TBD1 >
    [ User-Equipment-Info-IMEISV ]
    [ User-Equipment-Info-MAC ]
    [ User-Equipment-Info-EUI64 ]
    [ User-Equipment-Info-ModifiedEUI64 ]
    [ User-Equipment-Info-IMEI ]
    [ AVP ]
```

8.53. User-Equipment-Info-IMEISV AVP

The User-Equipment-Info-IMEISV (AVP Code TBD2) is of type OctetString. The User-Equipment-Info-IMEISV AVP contains the International Mobile Equipment Identifier and Software Version in the international IMEISV format according to 3GPP TS 23.003 [TGPPIMEI].

8.54. User-Equipment-Info-MAC AVP

The User-Equipment-Info-MAC (AVP Code TBD3) is of type OctetString. The User-Equipment-Info-MAC AVP contains the 48-bit MAC address is formatted as described in [RFC3580].

8.55. User-Equipment-Info-EUI64 AVP

The User-Equipment-Info-EUI64 (AVP Code TBD4) is of type OctetString. The UUser-Equipment-Info-EUI64 AVP contains the 64-bit identifier

used to identify the hardware instance of the product, as defined in [EUI64].

8.56. User-Equipment-Info-ModifiedEUI64 AVP

The User-Equipment-Info-ModifiedEUI64 (AVP Code TBD5) is of type OctetString. 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 [RFC4291] or by using some other methods referred to in the service-specific documentation. The User-Equipment-Info-ModifiedEUI64 AVP contains such identifiers.

8.57. User-Equipment-Info-IMEI AVP

The User-Equipment-Info-IMEI (AVP Code TBD6) is of type OctetString. The User-Equipment-Info-IMEI AVP contains the International Mobile Equipment Identifier in the international IMEI format according to 3GPP TS 23.003 [TGPPIMEI].

8.58. Subscription-Id-Extension AVP

The Subscription-Id-Extension AVP (AVP Code TBD7) is used to identify the end user's subscription and is of type Grouped. The Subscription-Id-Extension group AVP MUST include an AVP holding the subscription identifier. The type of this included AVP indicates the type of the subscription identifier. For each of the enumerated values of the Subscription-Id-Type AVP, there is a corresponding sub-AVP for use within the Subscription-Id-Extension group AVP. If a new identifier type is required a corresponding new sub-AVP SHOULD be defined for use within the Subscription-Id-Extension group AVP.

If full backward compatibility with [RFC4006] is required, then the Subscription-Id AVP MUST be used to indicate identifier types enumerated in the Subscription-Id-Type AVP, whereas the Subscription-Id-Extension AVP MUST be used only for newly defined identifier types. If full backward compatibility with [RFC4006] is not required, then the Subscription-Id-Extension AVP MAY be used to carry out the existing identifier types. In this case, Subscription-Id-Extension AVP MAY be sent together with Subscription-Id AVP.

Exactly one sub-AVP MUST be included inside the Subscription-Id-Extension AVP.

It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Subscription-Id-Extension ::= < AVP Header: TBD7 >
    [ Subscription-Id-E164 ]
    [ Subscription-Id-IMSI ]
    [ Subscription-Id-SIP-URI ]
    [ Subscription-Id-NAI ]
    [ Subscription-Id-Private ]
    [ AVP ]
```

8.59. Subscription-Id-E164 AVP

The Subscription-Id-E164 (AVP Code TBD8) is of type UTF8String. The Subscription-Id-E164 AVP contains the international E.164 format (e.g., MSISDN), according to the ITU-T E.164 numbering plan defined in [E164] and [CE164].

8.60. Subscription-Id-IMSI AVP

The Subscription-Id-IMSI (AVP Code TBD9) is of type UTF8String. The Subscription-Id-IMSI AVP contains the international IMSI format, according to the ITU-T E.212 numbering plan as defined in [E212] and [CE212].

8.61. Subscription-Id-SIP-URI AVP

The Subscription-Id-SIP-URI (AVP Code TBD10) is of type UTF8String. The Subscription-Id-SIP-URI AVP contains the identifier in the form of a SIP URI, as defined in [RFC3261].

8.62. Subscription-Id-NAI AVP

The Subscription-Id-NAI (AVP Code TBD11) is of type UTF8String. The Subscription-Id-NAI AVP contains the identifier in the form of a Network Access Identifier, as defined in [RFC7542].

8.63. Subscription-Id-Private AVP

The Subscription-Id-Private (AVP Code TBD12) is of type UTF8String. The Subscription-Id-Private AVP contains a credit-control server private identifier.

8.64. Redirect-Server-Extension AVP

The Redirect-Server-Extension AVP (AVP Code TBD13) 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 inside the QoS-Final-Unit-Indication AVP when the Final-Unit-

Action AVP is set to REDIRECT. If the type of the redirect server is one of the enumerated values of the Redirect-Address-Type AVP, then the credit-control server SHOULD send the information in the Redirect-Server AVP, in addition to or instead of the Redirect-Server-Extension AVP. This is in order to preserve backward compatibility with credit-control clients that support only [RFC4006]. Exactly one AVP MUST be included inside the Redirect-Server-Extension AVP.

It is defined as follows (per the grouped-avp-def of [RFC6733]):

```
Redirect-Server-Extension ::= < AVP Header: TBD13 >
    [ Redirect-Address-IPAddress ]
    [ Redirect-Address-URL ]
    [ Redirect-Address-SIP-URI ]
    [ AVP ]
```

8.65. Redirect-Address-IPAddress AVP

The Redirect-Address-IPAddress AVP (AVP Code TBD14) is of type Address and defines the IPv4 or IPv6 address of the redirect server with which the end user is to be connected when the account cannot cover the service cost.

When encoded as an IPv6 address in 16 bytes, the IPv4-mapped IPv6 format [RFC4291] MAY be used to indicate an IPv4 address.

8.66. Redirect-Address-URL AVP

The Redirect-Address-URL AVP (AVP Code TBD15) is of type UTF8String and defines the address of the redirect server with which the end user is to be connected when the account cannot cover the service cost. The address type is in the form of Uniform Resource Locator, as defined in [RFC1738].

8.67. Redirect-Address-SIP-URI AVP

The Redirect-Address-SIP-URI AVP (AVP Code TBD16) is of type UTF8String and defines the address of the redirect server with which the end user is to be connected when the account cannot cover the service cost. The address type is in the form of SIP Uniform Resource Identifier, as defined in [RFC3261].

8.68. QoS-Final-Unit-Indication AVP

The QoS-Final-Unit-Indication AVP (AVP Code TBD17) 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 QoS-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 QoS-Final-Unit-Indication AVP is included in a command.

If the Final-Unit-Action AVP is set to TERMINATE, the QoS-Final-Unit-Indication group MUST NOT contain any other AVPs.

If the Final-Unit-Action AVP is set to REDIRECT at least the Redirect-Server-Extension AVP MUST be present. The 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-Extension AVP or if the access to these services needs to be limited in some way (e.g., QoS).

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

The Filter-Rule AVP is defined in [RFC5777]. The Filter-Rule AVP can be used to define a specific condition and action combination. If used only with traffic conditions, it should define which traffic should be allowed when no more service units are granted. However, if QoS or treatment information exists in the AVP, these actions should be executed, e.g., limiting the allowed traffic with certain QoS.

When multiple Filter-Rule AVPs exist, precedence should be determined as defined in [RFC5777].

The Filter-Id AVP is defined in [RFC7155]. 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.

If the Final-Unit-Action AVP is set to TERMINATE, or set to RESTRICT_ACCESS and the action required is allow only traffic that could be classified using an IPFilterRule, or set to REDIRECT of a type which is one of the types in the Redirect-Address-Type AVP, then the credit-control server SHOULD send the information in the Final-Unit-Indication AVP, in addition to or instead of the QoS-Final-Unit-Indication AVP. This is in order to preserve backward compatibility with credit-control clients that support only [RFC4006].

The QoS-Final-Unit-Indication AVP is defined as follows (per the grouped-avp-def of [RFC6733]):

```
QoS-Final-Unit-Indication ::= < AVP Header: TBD17 >
    { Final-Unit-Action }
    *[ Filter-Rule ]
    *[ Filter-Id ]
    [ Redirect-Server-Extension ]
    *[ AVP ]
```

9. Result Code AVP Values

This section defines new Result-Code AVP [RFC6733] 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
QoS-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
Subscription-Id-Extension	0+	0
Termination-Cause	0-1	0
User-Equipment-Info	0-1	0
User-Equipment-Info-Extension	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 [RFC6733].

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 Figure 9, and interworking flow in Figure 10. 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.

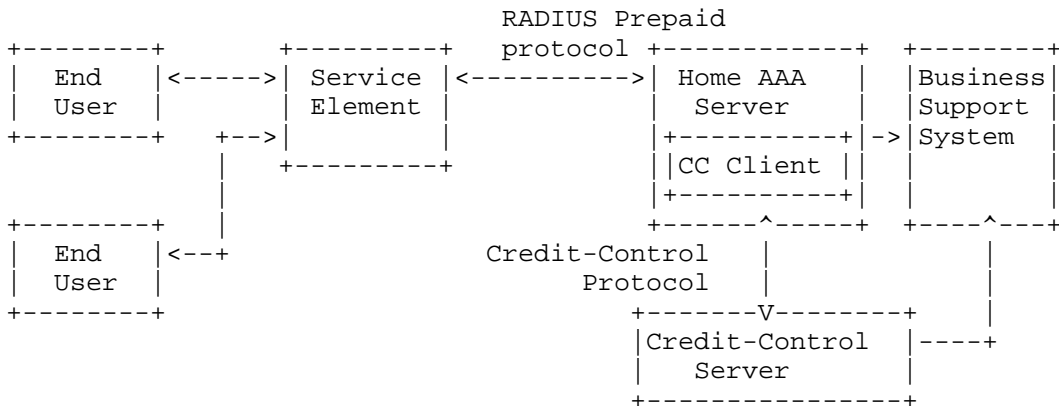


Figure 9: Credit-control architecture with service element containing translation agent, translating RADIUS prepaid to Diameter credit-control protocol

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.

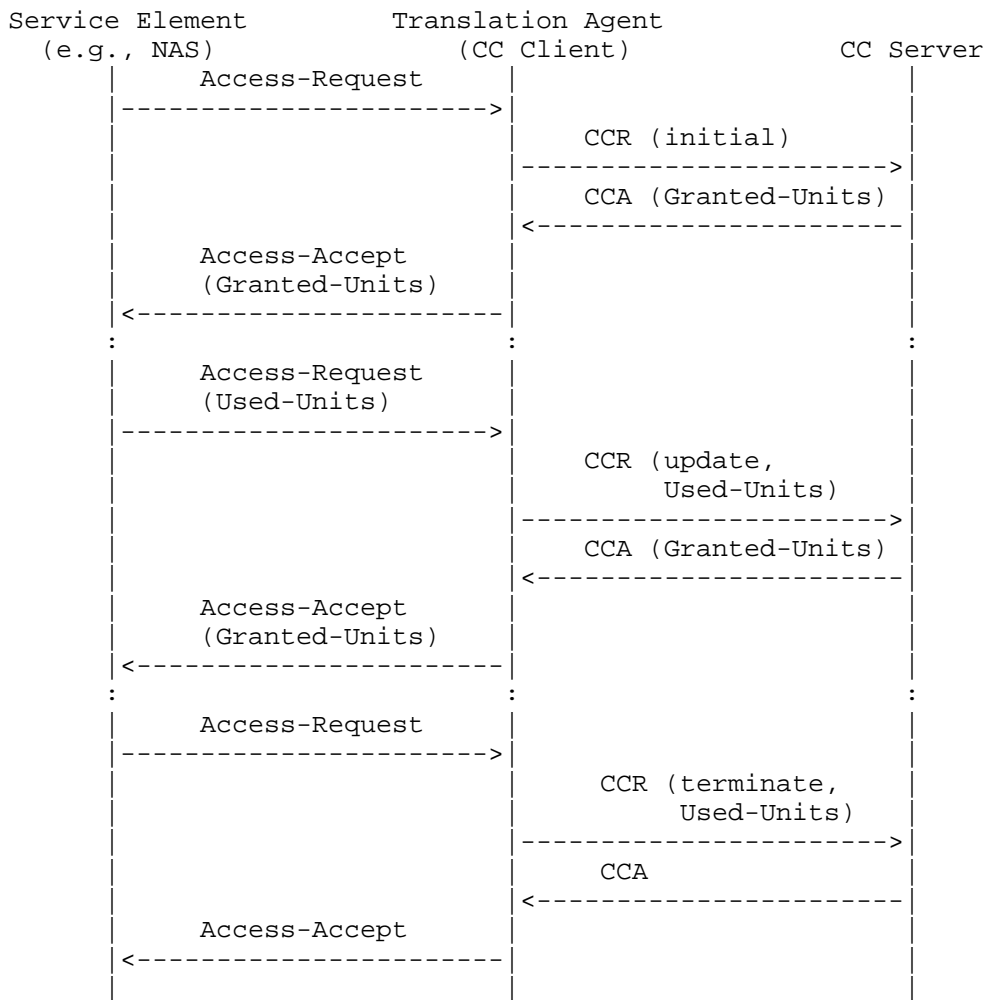


Figure 10: Message flow example with RADIUS prepaid - Diameter credit-control interworking

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 [RFC6733]. See Section 1.3 for more information.

12.2. Command Codes

This specification uses the value 272 from the Command code namespace defined in [RFC6733] 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 [RFC6733]. 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 [RFC6733]. 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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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 [RFC2434].

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

Security considerations regarding the Diameter protocol itself are discussed in [RFC6733]. Use of this application of Diameter MUST take into consideration the security issues and requirements of the base protocol.

This application includes a mechanism for application layer replay protection by means of the Session-Id from [RFC6733] 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/TCP, DTLS/SCTP or IPsec is sufficient. The details of TLS/TCP, DTLS/SCTP or IPsec related security considerations are discussed in the [RFC6733].

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 [RFC6733], 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

- [CE164] "Complement to ITU-T Recommendation E.164 (05/1997):"List of ITU-T Recommendation E.164 assigned country codes"", June 2000.

- [CE212] "Complement to ITU-T Recommendation E.212 (11/1997):" List of mobile country or geographical area codes", February 1999.
- [E164] "Recommendation E.164/I.331 (05/97): The International Public Telecommunication Numbering Plan.", 1997.
- [E212] "Recommendation E.212 (11/98): The international identification plan for mobile terminals and mobile users.", 1998.
- [EUI64] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", March 1997, <<http://standards.ieee.org/regauth/oui/tutorials/EUI64.html>>.
- [ISO4217] "Codes for the representation of currencies and funds, International Standard ISO 4217", 2001.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [RFC1738] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, DOI 10.17487/RFC1738, December 1994, <<http://www.rfc-editor.org/info/rfc1738>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, DOI 10.17487/RFC2434, October 1998, <<http://www.rfc-editor.org/info/rfc2434>>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <<http://www.rfc-editor.org/info/rfc3261>>.
- [RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, DOI 10.17487/RFC3539, June 2003, <<http://www.rfc-editor.org/info/rfc3539>>.

- [RFC3580] 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, DOI 10.17487/RFC3580, September 2003, <<http://www.rfc-editor.org/info/rfc3580>>.
- [RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, "Diameter Credit-Control Application", RFC 4006, DOI 10.17487/RFC4006, August 2005, <<http://www.rfc-editor.org/info/rfc4006>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., Ed., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, DOI 10.17487/RFC5777, February 2010, <<http://www.rfc-editor.org/info/rfc5777>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<http://www.rfc-editor.org/info/rfc5952>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.
- [RFC7155] Zorn, G., Ed., "Diameter Network Access Server Application", RFC 7155, DOI 10.17487/RFC7155, April 2014, <<http://www.rfc-editor.org/info/rfc7155>>.
- [RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <<http://www.rfc-editor.org/info/rfc7542>>.
- [TGPPCHARG] 3rd Generation Partnership Project, "Technical Specification Group Services and System Aspects, Service aspects; Charging and Billing, (release 13), 3GPP TS 22.115 v. 13.3.0", 2016-03.

[TGPPIMEI]

3rd Generation Partnership Project, "Technical Specification Group Core Network, Numbering, addressing and identification, (release 13), 3GPP TS 23.003 v. 13.5.0", 2016-04.

15.2. Informative References

- [DIAMEAP] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", Work in Progress.
- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, DOI 10.17487/RFC2866, June 2000, <<http://www.rfc-editor.org/info/rfc2866>>.
- [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, DOI 10.17487/RFC3725, April 2004, <<http://www.rfc-editor.org/info/rfc3725>>.
- [RFC4004] Calhoun, P., Johansson, T., Perkins, C., Hiller, T., Ed., and P. McCann, "Diameter Mobile IPv4 Application", RFC 4004, DOI 10.17487/RFC4004, August 2005, <<http://www.rfc-editor.org/info/rfc4004>>.

Appendix A. Acknowledgements

The original authors of RFC4006 are: Harri Hakala, Leena Mattila, Juha-Pekka Koskinen, Marco Stura, and John Loughney.

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 B. Credit-Control Sequences

B.1. Flow I

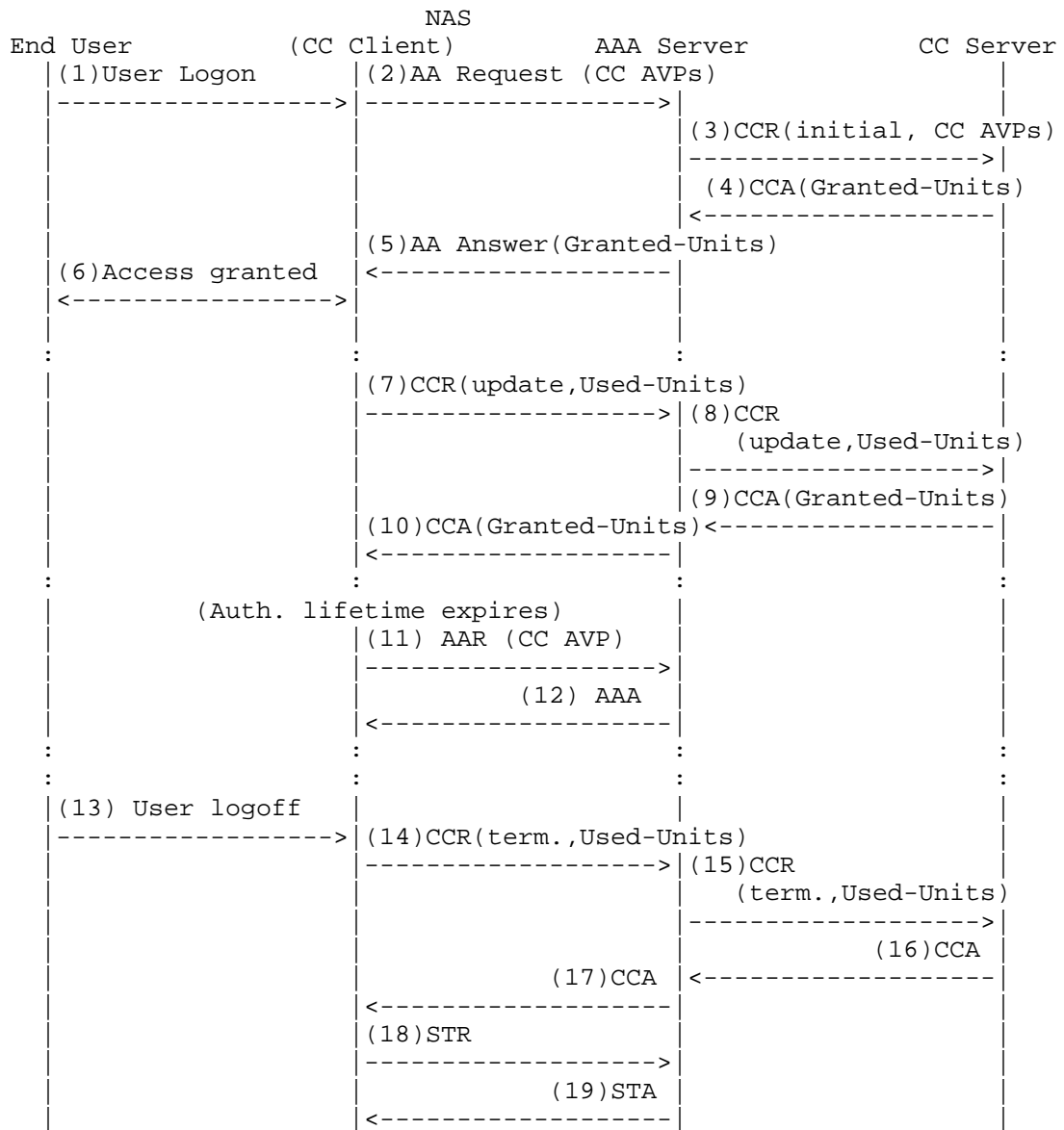


Figure 11: Flow I

A credit-control flow for Network Access Services prepaid is shown in Figure 11. The Diameter [RFC7155] 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 [RFC7155]. 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).

B.2. Flow II

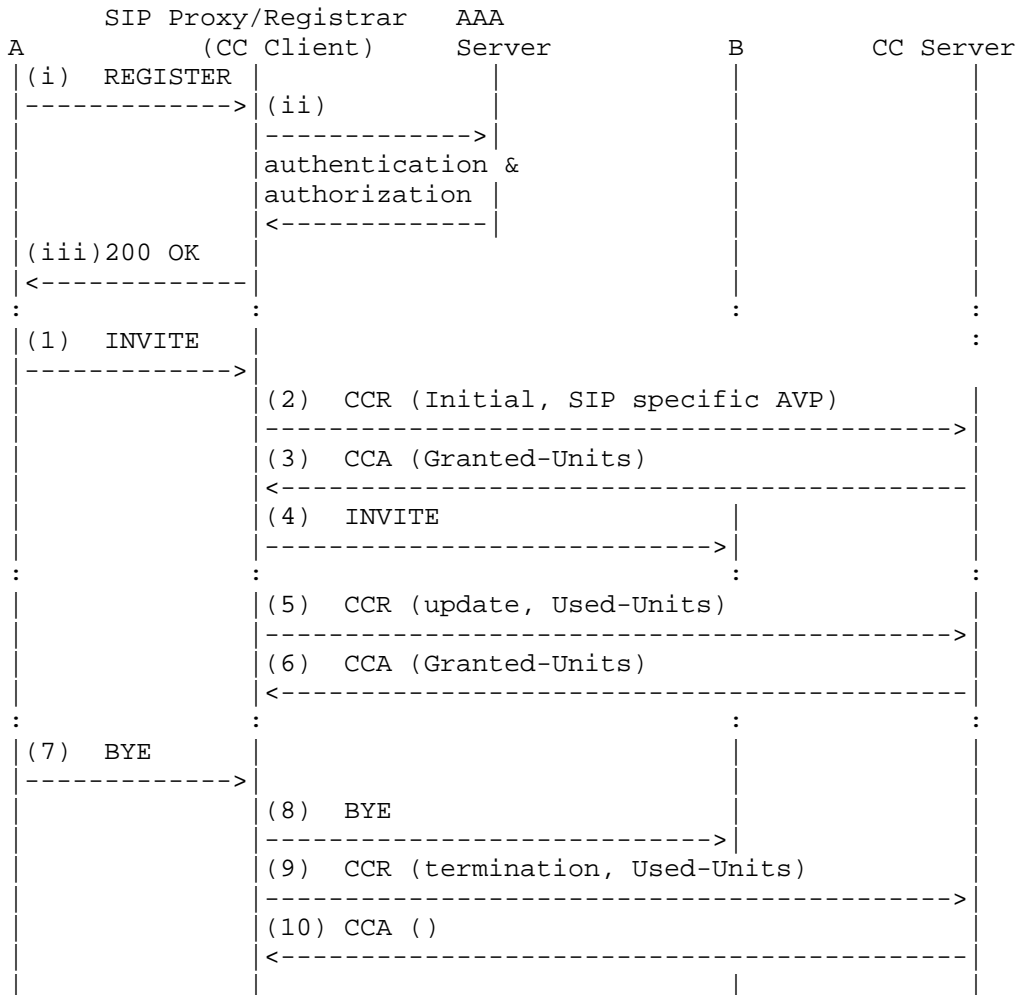


Figure 12: Flow II

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 [RFC3261]). 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).

B.3. Flow III

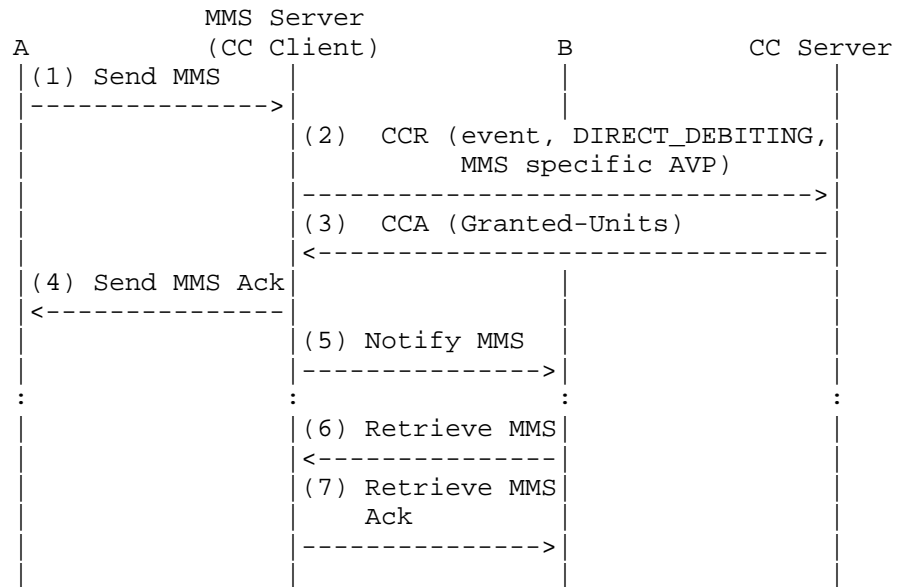


Figure 13: Flow III

A credit-control flow for Multimedia Messaging Services is shown in Figure 13. 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).

B.4. Flow IV

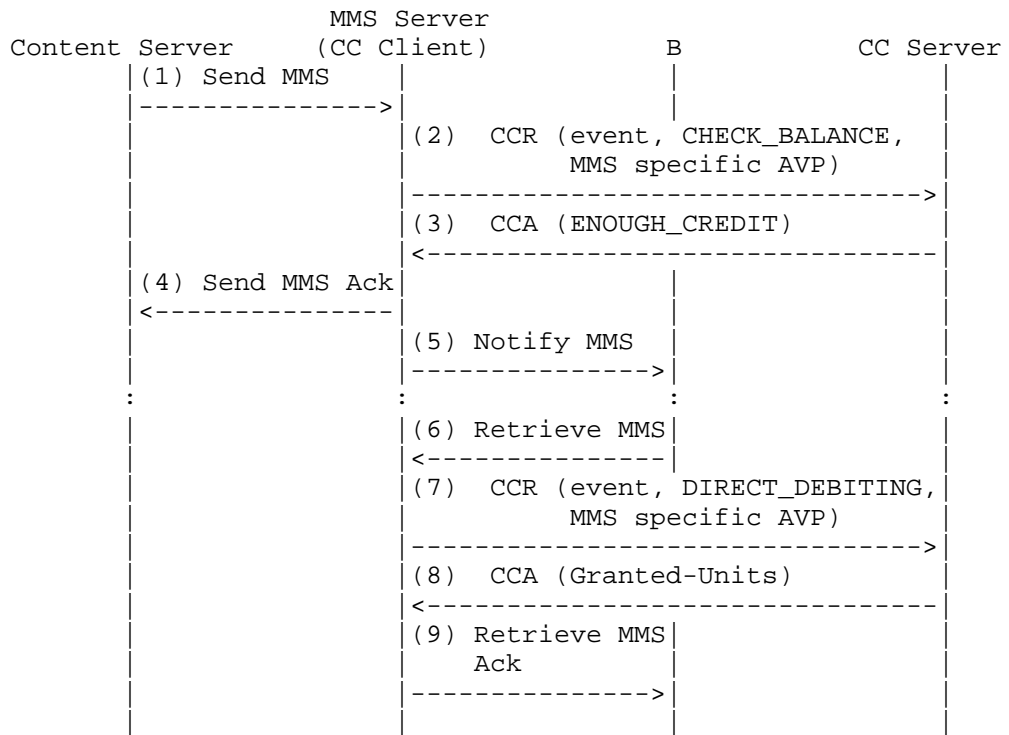


Figure 14: Flow IV

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 14. 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.

B.5. Flow V

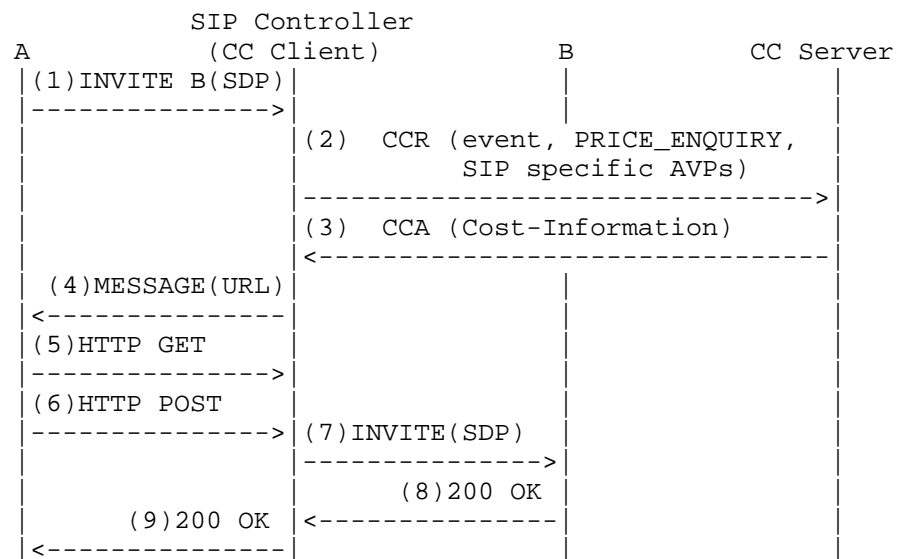


Figure 15: Flow V

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 15 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).

B.6. Flow VI

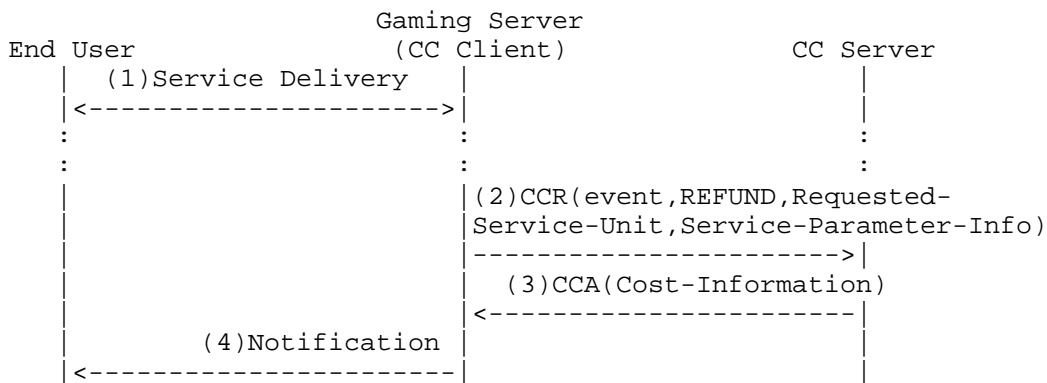


Figure 16: Flow VI

Figure 16 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 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).

B.7. Flow VII

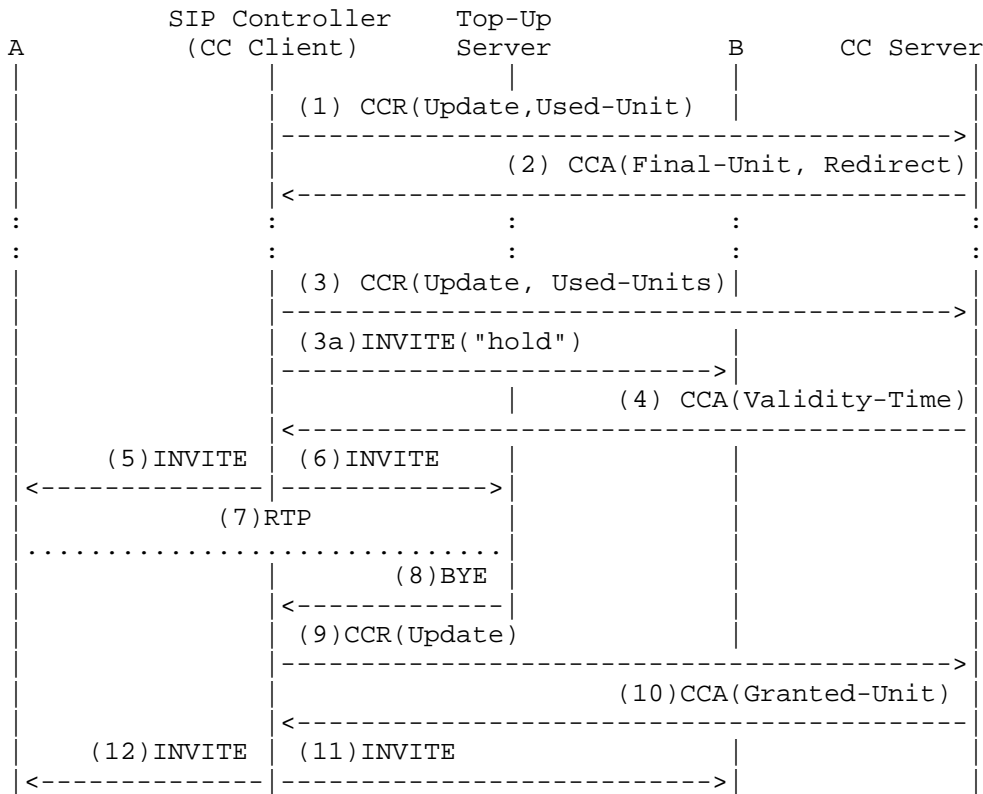


Figure 17: Flow VII

Figure 17 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).

B.8. Flow VIII

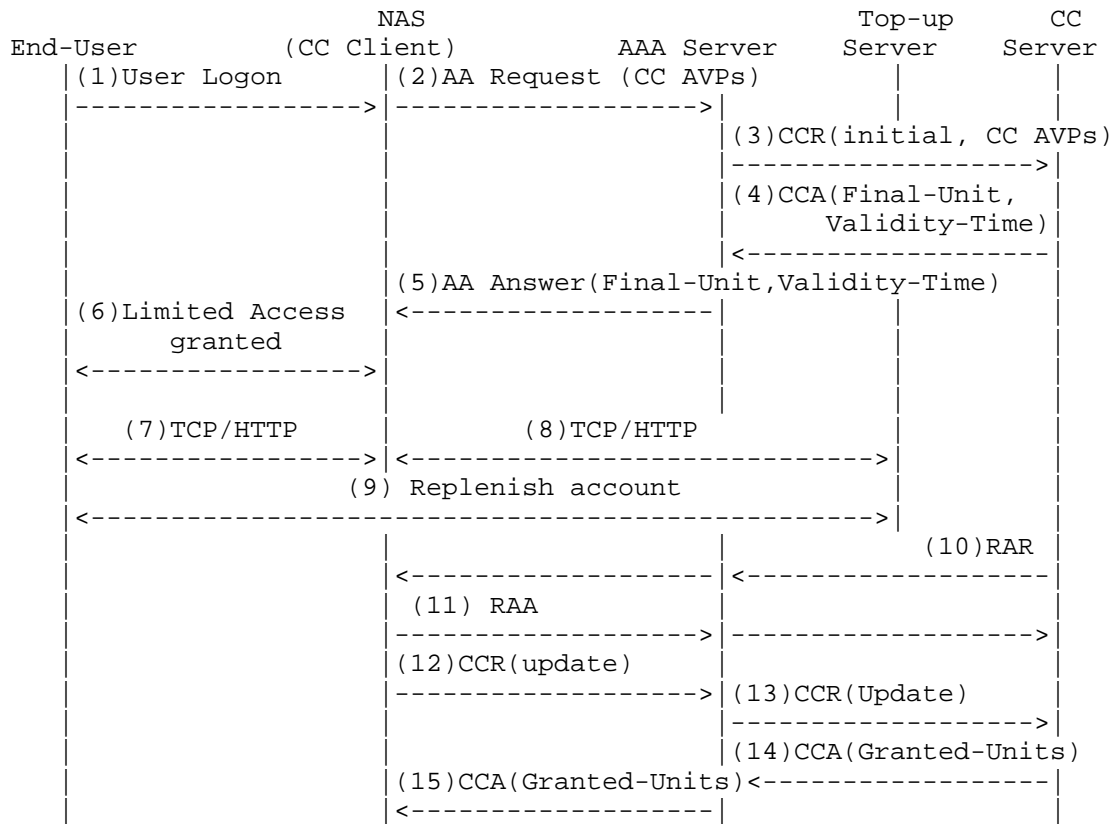


Figure 18: Flow VIII

Figure 18 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 [RFC7155] 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 [RFC7155]. 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

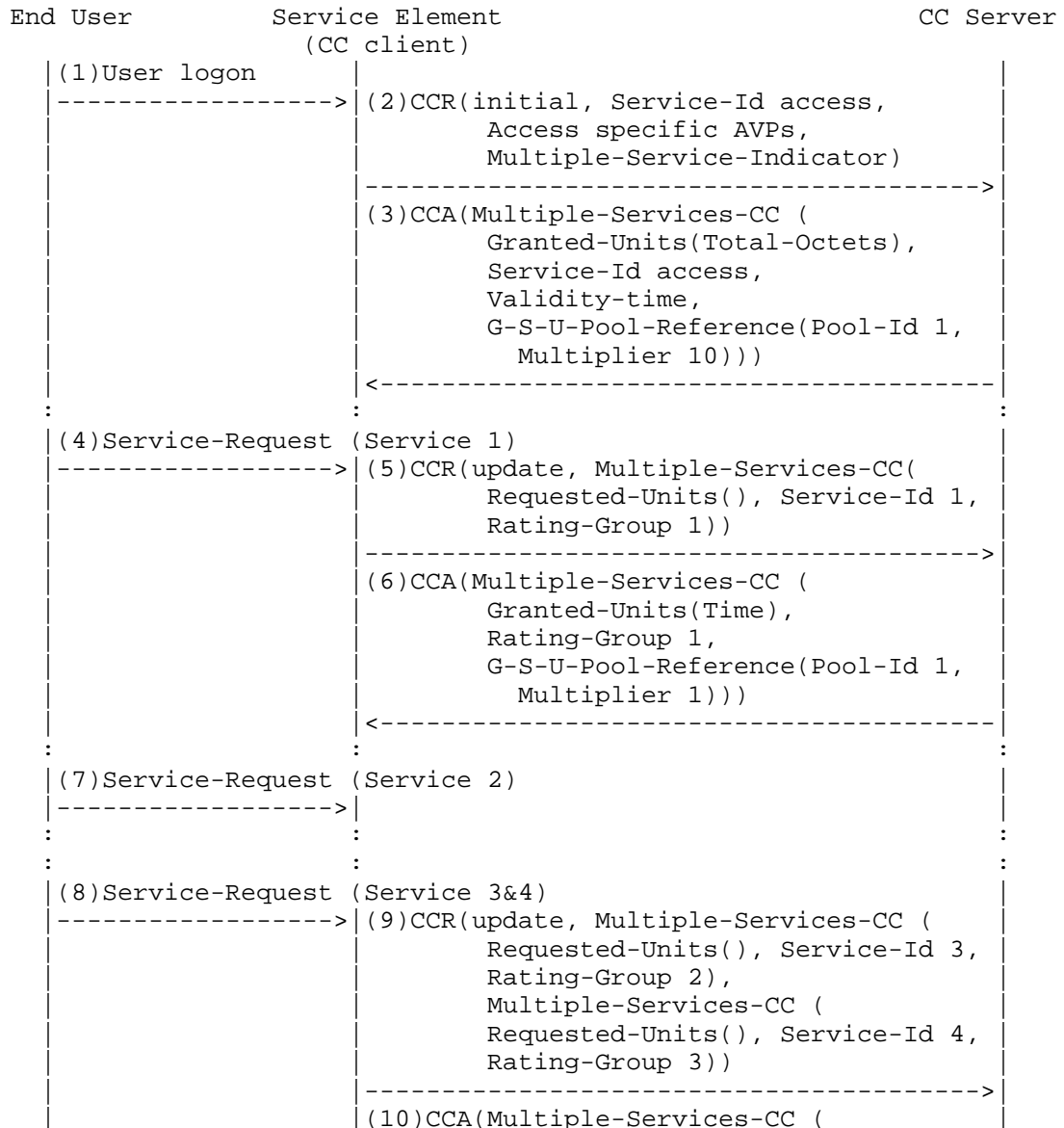
establish a credit-control session. (The home Diameter AAA server may forward service specific AVPs received from the NAS as input for the rating process.) The Diameter credit-control server checks the end user's account balance, determines that the account cannot cover the cost of the service, and initiates 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.

B.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.




```

----->| (16)CCR(term,
         |     Multiple-Services-CC (
         |         Used-Units(In-Octets,Out-Octets),
         |         Service-Id access),
         |     Multiple-Services-CC (
         |         Used-Units(Time),
         |         Service-Id 1, Rating-Group 1),
         |     Multiple-Services-CC (
         |         Used-Units(Time),
         |         Service-Id 2, Rating-Group 1))
         |----->
         | (17)CCA(term)
         |<-----

```

Figure 19: 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 to 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 to 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).

Appendix C. Changes relative to RFC4006

The following changes were made relative to RFC4006:

- Update references to obsolete RFC 3588 to refer to RFC 6733.
- Update references to obsolete RFC 4005 to refer to RFC 7155.
- Update references to obsolete RFC 2486 to refer to RFC 7542.
- Update references to current 3GPP documents.
- Update AVP per Errata ID 3329.
- Update reference to "IPsec or TLS" to be "TLS/TCP, DTLS/SCTP or IPsec".

Clarify Filter-Rule AVP in Restrict Access Action.

Remove Encr column from AVP flag rules.

Clarify that RESTRICT_ACCESS action applies after consumption of final granted units (Section 5.6.3).

Clarify that values in Used-Service-Unit AVP may exceed Granted-Service-Unit AVP (Section 8.19).

Clarify that IPv6 representation in Redirect-Address-Type AVP conforms to RFC5952 (Section 8.38).

Describe immediate graceful service termination procedure (in Section 5.6).

Add extensible User-Equipment-Info-Extension AVP and included types (from Section 8.52 to Section 8.57).

Add extensible Subscription-Id-Extension AVP and included types (from Section 8.58 to Section 8.63).

Add extensible Redirect-Server-Extension AVP and included types (from Section 8.64 to Section 8.67).

Add extensible QoS-Final-Unit-Indication AVP (in Section 8.68).

Updated Security Section to include language consistent with structures of latest base protocol specification.

Authors' Addresses

Lyle Bertz (editor)
Sprint
6220 Sprint Parkway
Overland Park, KS 66251
United States

Email: lyleb551144@gmail.com

David Dolson (editor)
Sandvine
408 Albert Street
Waterloo, ON N2L 3V3
Canada

Phone: +1 519 880 2400
Email: ddolson@sandvine.com

Yuval Lifshitz (editor)
Sandvine
408 Albert Street
Waterloo, ON N2L 3V3
Canada

Phone: +1 519 880 2400
Email: ylifshitz@sandvine.com