

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: January 4, 2015

S. Donovan  
B. Campbell  
Oracle  
July 3, 2014

Analysis of Agent Use Cases for Diameter Overload Information Conveyance  
(DOIC)  
draft-donovan-doic-agent-cases-00

Abstract

The Diameter Overload Information Conveyance (DOIC) solution describes a mechanism for exchanging information about Diameter Overload among Diameter nodes. A DOIC node is a Diameter node that acts as either a reporting node or a reacting node. A reporting node originates overload reports, requesting reacting nodes to reduce the amount of traffic sent. DOIC allows Diameter agents to act as reporting nodes, reacting nodes, or both, but does not describe agent behavior. This document explores several use cases for agents to participate in overload control, and makes recommendations for certain agent behaviors to be added to DOIC.

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 January 4, 2015.

Copyright Notice

Copyright (c) 2014 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 . . . . .	2
1.1. Terminology and Abbreviations . . . . .	3
2. Deployment Architectures . . . . .	4
3. Diameter Routing . . . . .	5
4. Overload Abatement Methods . . . . .	6
5. DOIC Use Cases . . . . .	7
5.1. Simple Agent . . . . .	9
5.1.1. Capability Announcement . . . . .	9
5.1.2. Overload Report Handling . . . . .	11
5.1.3. DOIC Specification Impacts . . . . .	17
5.2. Mixed Capabilities . . . . .	17
5.2.1. Capability Announcement . . . . .	17
5.2.2. Mixed Abatement Algorithms . . . . .	18
5.2.3. DOIC Specification Impacts . . . . .	20
5.3. Non-Supporting Nodes . . . . .	20
5.3.1. Non-Supporting Transaction Client . . . . .	20
5.3.2. Non-supporting Transaction Server . . . . .	24
5.3.3. Non-Supporting Agent . . . . .	26
5.3.4. DOIC Specification Impacts . . . . .	27
5.4. Inter Domain Authentication . . . . .	27
5.4.1. DOIC Specification Impacts . . . . .	29
6. Recommendations . . . . .	29
6.1. General Recommendations . . . . .	30
6.2. Agent Behavior Recommendations . . . . .	30
6.2.1. Capabilites Exchange Behaviors . . . . .	30
6.2.2. Overload Report Behaviors . . . . .	31
7. Security Considerations . . . . .	32
8. References . . . . .	33
8.1. Normative References . . . . .	33
8.2. Informative References . . . . .	33
Authors' Addresses . . . . .	34

## 1. Introduction

The Diameter Overload Information Conveyance (DOIC) [I-D.ietf-dime-ovli] solution describes a mechanism for exchanging diameter overload information among Diameter nodes. DOIC defines the concept of a DOIC endpoint (referred to as a "DOIC node" in this document.) A DOIC node is a Diameter node that acts as either a

reporting node or a reacting node. A reporting node originates overload reports, requesting reacting nodes to reduce the offered load. An overload report has a "type". The type of overload report determines the scope of the request for traffic reduction, and possibly other semantics.

DOIC nodes do not necessarily correspond to Diameter clients and servers. Any Diameter node that supports DOIC is a DOIC node. This includes Diameter agents, as well as Diameter clients and servers. However, DOIC does not currently describe how agents should behave as part of an overload control solution. This document explores several use cases for agents to participate in overload control, and makes recommendations for certain agent behaviors to be added to DOIC.

### 1.1. Terminology and Abbreviations

Diameter Node: A Diameter client, agent or server.

DOIC node: A Diameter node that supports DOIC. A DOIC node can simultaneously be both a reacting node and a reporting node.

Reacting node: A DOIC node that can receive overload reports from a reporting node and ensures that the overload reports are honored.

Reporting node: A DOIC node that can send overload reports, requesting overload abatement.

Abating node: A reacting node that directly performs overload abatement.

Transaction Client (TC) A diameter node that originates a Diameter request. This is distinct from "Diameter Client" as used in RFC 6733. Note that a Diameter Server acts as a TC if and when it originates a request towards a Diameter Client.

TransactionServer (TS) A diameter node that consumes a Diameter request, and responds with a Diameter answer. This is distinct from "Diameter Server" as used in RFC 6733. Note that a Diameter Client acts as a TS if and when it answers a request sent by a Diameter Server.

Client Application The application that uses Diameter for various Authentication, Authorization, and/or Accounting (AAA) functions. For example, a Network Access Server (NAS) performs certain network attachment services, detachment services, packet forwarding, etc. These are collectively the NAS client application, which depends on Diameter for AAA services.

**Overload Abatement:** Actions taken by a reacting node to reduce the load offered to an overloaded Diameter node. The specific actions required to perform overload abatement are described by the DOIC algorithm. Overload abatement actions may involve local traffic reduction, or delegation of actions towards the client. Local traffic reduction can be achieved by either throttling a request or routing a request to a different destination.

**Throttling** Overload abatement through the rejection of some number of requests. Throttling at an agent requires the agent to reject requests with appropriate error codes. Throttling at a transaction client requires the client to indicate appropriate errors to the client application

**Diversion** Overload abatement through the routing of some number of requests away from an overloaded node towards one or more appropriate nodes that are less-overloaded.

## 2. Deployment Architectures

This section outlines the deployment architectures used to determine agent related Diameter DOIC requirements.

These deployment architectures include the use of Diameter agents to route Diameter requests between Diameter clients and Diameter servers.

In all cases, a small number of client and server nodes are shown for simplicity. Adding additional clients and/or servers does not change the fundamental characteristics of the deployments.

Figure 1 shows an architecture with a single agent sitting between Diameter clients and Diameter servers.

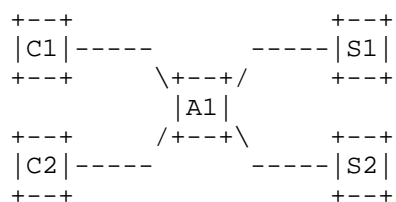


Figure 1

Figure 2 shows an architecture with multiple agents sitting between Diameter clients and Diameter servers.

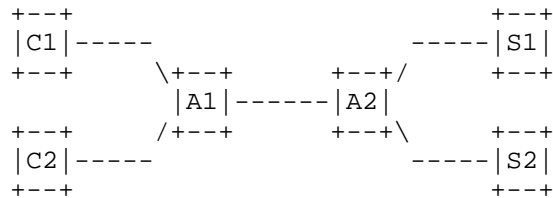


Figure 2

This document focuses on use cases that involve agents, and does not therefore include scenarios with no agent(s) between the transaction client and transaction server. It is, however, important that any changes to the DOIC solution introduced to support networks that contain agents also work when there is no agent sitting between Diameter clients and servers.

### 3. Diameter Routing

Diameter supports two primary methods [RFC6733] for nodes to select the next hop for a request. Normally, Diameter nodes base peer-selection on the Destination-Realm and Application-Id AVP values. That is, they select a next hop from their Diameter peer table entry that matches the realm and application of the request. For the sake of this analysis, we refer to requests routed by this method as "realm-routed requests".

A Diameter TC may also control the final destination of a request by inserting a Destination-Host AVP. When a node forwards a request that includes Destination-Host, it checks to see if it has a matching Diameter identity in its peer table. If so, it forwards the request to that peer. Otherwise, it follows the normal peer-selection for the realm and application. We refer to requests routed this way as "host-routed requests".

In general, throttling (Section 4) is the the only abatement technique that works for host-routed requests. Diversion (Section 4) is typically not possible, since only a single TS can handle any given request.

There may be some exceptions to this rule. For example, a node might have multiple peer table entries that share the same

Diameter Identity. A node might map Diameter identities in a way that results in multiple next-hop destinations for a given Destination-Host value.

On the other hand, diversion is often useful for abating realm-routed traffic. Since realm-routed requests are not bound to a particular TS, it is often possible to divert a number of them to other servers that are less overloaded.

#### 4. Overload Abatement Methods

When a Diameter node becomes overloaded, there often must be a reduction of the number of both realm-routed and host-routed requests, in order to have the desired reduction of the overall offered-load. We refer to this reductions as "Overload Abatement".

There are two ways to perform overload abatement. The first is to reject some number of Diameter requests, also known as "throttling". When a TC throttles traffic, it rejects or defers certain client application requests, as appropriate for the client application. When an agent or TS throttles traffic, it rejects Diameter requests with an appropriate error code, so that the TC can behave correctly at the client application layer.

The second way to abate overload is to move some number of requests from an overloaded node to one or more eligible nodes that are less overloaded. For the purposes of this draft, we refer to this abatement method as "Diversion".

Either method, separately or in combination, continue over the duration of the overload condition.

There are a few architectural principles that should be considered when building Diameter networks to be resilient to overload, or when deploying DOIC into existing Diameter network.

All things being equal, diversion is better than throttling. Diversion potentially allows more requests to succeed, which will almost always have less negative impact on the client application. However, there are situations where diversion is not possible. For example, diversion is usually not possible for host-routed requests (see Section 3). Diversion may not be helpful if all potential destinations are overloaded. If proprietary load balancing mechanisms are in use, diversion for DOIC purposes may be redundant with those mechanisms.

If diversion is performed, the diversion should occur as close as possible to the TS, but not at the TS itself (since this would defeat

the point of overload abatement.) Diversion should optimally occur at an immediate peer to the TS; that is, a node that shares a direct transport connection with the TS. A directly connected peer will have the most knowledge of alternative destinations and their current loads. If nodes further from the TS attempt to diversion, topology knowledge and overload state knowledge must be pushed further down the chain. Even then, a node usually cannot control how a realm-routed request might be routed upstream of the immediate peer.

Throttling should occur at, or as close as possible, to the TC. The TC has the best knowledge of the client application and its current state. The TC can choose to reject requests that have the least impact on the client application, or provide the most effect for traffic reduction over time. Furthermore, throttling at the TC completely avoids Diameter overhead for rejected requests. Each additional hop traversed by requests that will eventually be rejected increases the impact of those requests.

## 5. DOIC Use Cases

This section outlines example use cases involving agents. Each of these use cases is evaluated with the goal of identifying any required changes to [I-D.ietf-dime-ovli] needed to support the use case.

The following is the list of use cases considered. This is not an exhaustive list of DOIC use cases but is rather a list of use cases identified by the authors as being impacted by the presence of agents in the deployment.

- o Simple Agent - Overload capability announcement and overload report handling in a deployment with a single agent as illustrated in Figure 1. In this case all Diameter nodes are assumed to support DOIC. This use case is discussed in Section 5.1.

This use case includes four sub-cases:

1. OC Capability Announcement where the TC and Agent support the same OC capabilities.
2. Host overload report handling for host-routed requests. This case illustrates throttling of host-routed requests at the transaction client and throttling of realm-routed requests at the agent.
3. Host overload report handling realm-routed requests when there is a second TS to which requests can be diverted when one of the servers is in an overload state.

4. Multiple host overload reports resulting in a realm overload report.

- o Mixed Capabilities - Overload capability announcement and overload report handling in deployments where agents support capabilities that are not included in the set of capabilities advertised by reacting nodes. This use case is discussed in Section 5.2.

This use case illustrates one scenario where an agent consumes an overload report and replaces it with a new overload report of a different type.

- o Non-supporting DOIC nodes - Agent behavior in the face of Diameter nodes that do not support the DOIC solution. These use cases are addressed in Section 5.3. There are four sub-use cases that are addressed:

- \* Non-supporting TC. In this case a DOIC supporting agent handles overload abatement on behalf of the non-supporting TC. An agent or a reporting node can detect if there is a reacting node in the path of a request by the presence of the OC-Supported-Features AVP in the request message. This use case is discussed in Section 5.3.1.

- \* Non-supporting TS. In this case a DOIC supporting agent may act as the reporting node on behalf of the TS. In this case a DOIC supporting agent can detect if there is a reporting node in the path of the transaction by the presence of the OC-Supported-Features AVP in the answer message for the transaction. This use case is discussed in Section 5.3.2.

- \* Non-supporting agent between the TC and a supporting agent.

- \* Non-supporting agent between a supporting agent and the TS. In this case, the agent that supports DOIC cannot reliably divert requests as a result of a host report. This use case is discussed in Section 5.3.3.

This use case illustrates when this deployment scenario is not recommended.

- o Inter domain or untrusted node authorization.

This use case illustrates one case where a node needs to know if an OC-S-F AVP came from a supported peer, or was forwarded by a non-supporting peer. This use case is discussed in Section 5.4.



### 5.1. Simple Agent

This section addresses overload capability announcement and overload report handling in a deployment with a single agent as illustrated in Figure 1.

This use case assumes that all nodes support DOIC and that all nodes support the same set of overload features.

This use case includes four sub-cases:

1. OC Capability Announcement where the TC and Agent support the same OC capabilities.
2. Host overload report handling for host-routed requests. This case illustrates throttling of host-routed requests at the transaction client and throttling of realm-routed requests at the agent.
3. Host overload report handling realm-routed requests when there is a second TS to which requests can be diverted when one of the servers is in an overload state.
4. Multiple host overload reports resulting in a realm overload report.

#### 5.1.1. Capability Announcement

This section explores capability announcement for the simple agent use case.

This use case assumes that the capabilities supported by the TC and those supported by the agent are the same. (A scenario with differing capabilities is supported in discussed in Section 5.2.)

Figure 3 shows the message flow for this use case.

The nomenclature OC-S-F:x is short for OC-Supported-Feature with the ":x" indicating the Diameter node that inserted the AVP into the message.

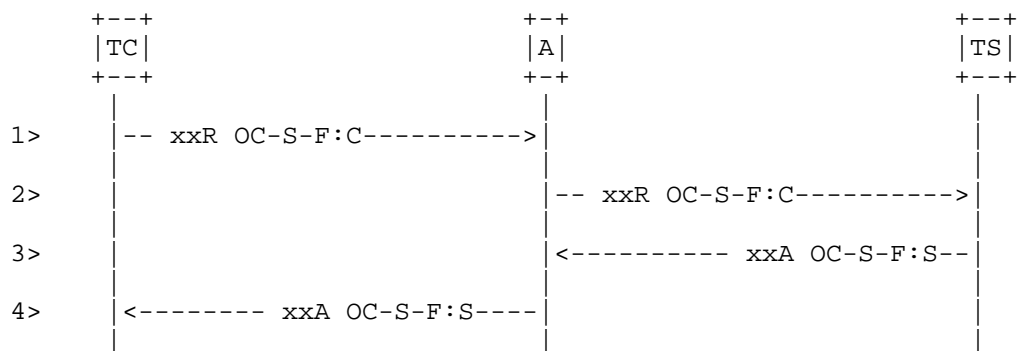


Figure 3

1. The transaction client (TC) originates a request. The TC supports DOIC and, as such, includes the OC-Supported-Features AVP in all requests. The OC-S-F AVP contains the clients capabilities.
2. The agent inspects the OC-S-F AVP and determines that the agent supports the same set of OC features. The agent relays the request unchanged to the server.

Note: It is an open question whether the agent needs to include an indication that it also supports DOIC or if attribution of the OC-S-F is needed. One could also question whether the agent forwarded OC-S-F:C unchanged, rather than consuming the OC-S-F, and inserted another identical one. This is likely a purely philosophical difference, but might impact the inter-domain authorization use case (Section 5.4).

3. The transaction server (TS), acting as the reporting node, inspects the OC-S-F AVP in the request and generates an OC-S-F to be included in the answer message. This is done according to the behavior defined in the DOIC specification [I-D.ietf-dime-ovli].
4. The agent relays the answer message unchanged.

The presence of the OC-S-F header in the answer message indicates to the TC that it needs to be prepared for overload reports in subsequent requests of the same type.

With the loss algorithm defined in [I-D.ietf-dime-ovli] there is no explicit action required of the TC. Stateful abatement algorithms will likely require action to be taken by the TC to be able to handle subsequent overload reports.

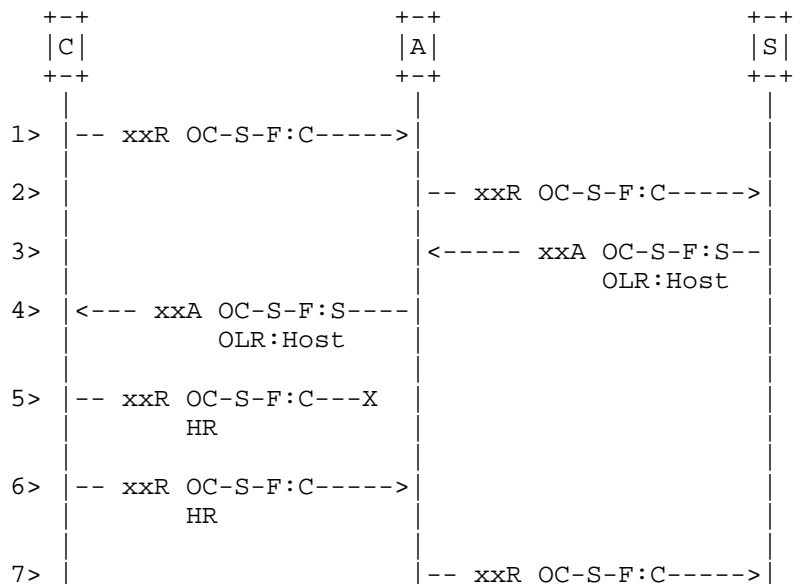
### 5.1.2. Overload Report Handling

This section addresses overload report handling in a deployment with a single agent as illustrated in Figure 1.

The following three scenarios are illustrated:

- o Figure 4 shows a message flow illustrating handling of host reports for host-routed requests and realm-routed requests when there is a single TS.
- o Figure 5 shows the handling of realm-routed requests when there is a second TS to which requests can be diverted when one of the servers is in an overload state.
- o Figure 6 illustrates the agents behavior when it has received a host report from all servers. In this case the agent generates a Realm report. This is only possible when the agent knows the overload state of all TSs for the given realm and application.

In these message flows, "xxR" indicates a Diameter request, and "xxA" indicates an answer. "HR" under a request indicates that the request is host-routed. "RR" indicates the request is realm-routed. If neither is present for a request, then it can be either host or realm routed. "OLR:Host" indicates an overload report of type Host. "OLR:Realm" indicates an overload report of type Realm.



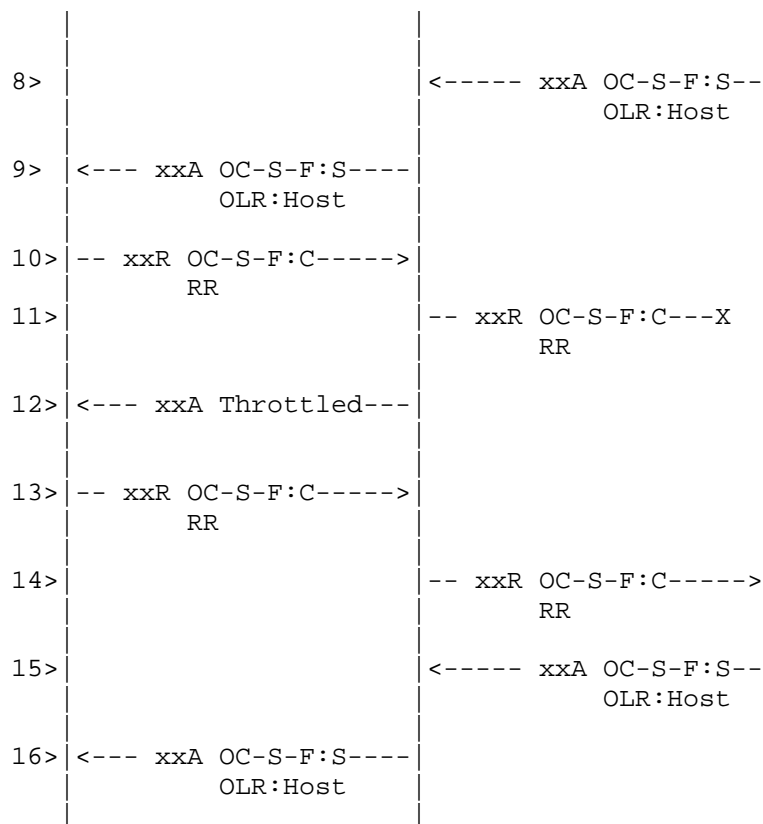


Figure 4

1. Same as in Figure 3.
2. Same as in Figure 3.
3. S, acting as a reporting node, has determined that it needs to request a reduction in traffic. S includes the OC-S-F AVP per [I-D.ietf-dime-ovli], and selects the loss algorithm for the included report. S also includes the OC-OLR AVP to indicate the requested reduction in traffic.
4. A saves the overload state of S based on the OC-OLR AVP. A will use this overload state for handling of future realm routed requests.

This behavior is not yet specified in the DOIC specification. It is based on the principle that only nodes with a direct

transport connection to an overloaded host should throttle those requests as other nodes earlier in the requests path do not have the topology knowledge to know if diversion of the request would have been successful.

A relays the answer message without change to OC-S-F or OC-OLR. Upon receipt of the answer, C saves overload state based on the overload report.

5. C invokes the "loss" algorithm on host-routed requests. This step illustrates a host-routed request that is rejected locally by C due to throttling. C gives application appropriate feedback to the client application.
6. This step represents a Host-routed requests that survived abatement. Such requests are handled the same as if there where no overload report for the host to which the request is routed.
7. A relays the request based on the included Destination-Host AVP.
8. A generates an answer which includes the OC-S-F AVP and OC-OLR AVP.
9. If the OC-OLR is new, then A updates the overload state associated with the report. A relays the answer without change to OC-S-F or OC-OLR.  
  
C determines if the OC-OLR is new. If so, C updates its locally stored overload state for S.
10. C originates a realm-routed request. C does not apply abatement to this request since it does not match any locally stored overload state (in this scenario, a realm overload report has not yet been sent.)
11. A determines that there is overload state associated with this request (the host report received from S). A uses this overload state as input to routing decisions for the request. In this case it is assumed that there is no alternative route to divert request toward and, as such, A applies throttling, and rejects the request.
12. A generates an error response indicating that the request was throttled and should not be retried.
13. C originates another realm-routed request.

14. A determines that there is overload state associated with this request (the host report received from S). A uses this overload state as input to routing decisions for the request. This request survives abatement and is routed to S.
15. S generates an answer message.
16. A relays the answer.

The following shows the case of host overload report handling of realm-routed requests when there is a second TS to which requests can be diverted when one of the servers is in an overload state.

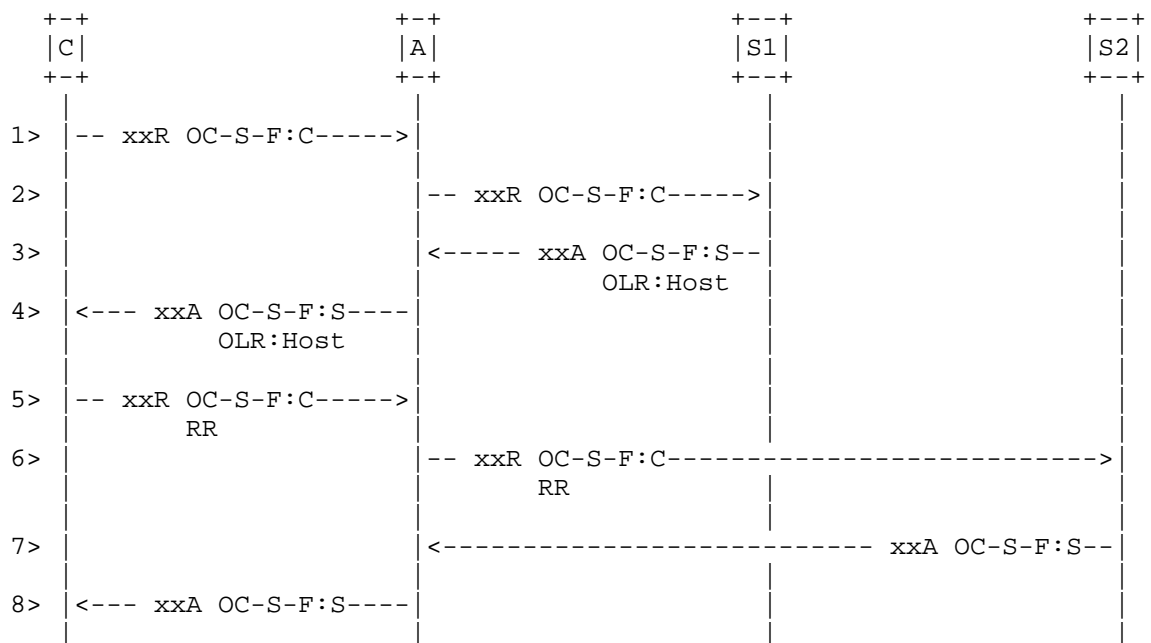


Figure 5

1. Same as in Figure 3.
2. Same as in Figure 3.
3. Same as in Figure 4.
4. Same as in Figure 4.

5. C originates a realm-routed request. C does not apply overload abatement to this request as it does not match any locally stored overload state (the assumption for this scenario is that a realm overload report has not yet been sent.)
6. A determines that there is overload state associated with this request (the host report received from S1). A uses this overload state as input to routing decisions for the request. In this case, it is assumed that the request would have been normally routed to S1 but is instead routed to S2 as a result of the overload report.
7. S2 generates an answer message.
8. A relays the answer.

The following illustrates the agents behavior when it has received a host report from all servers. In this case the agent generates a Realm report. This is only possible when the agent knows the overload state of all TSs for the given realm and application.

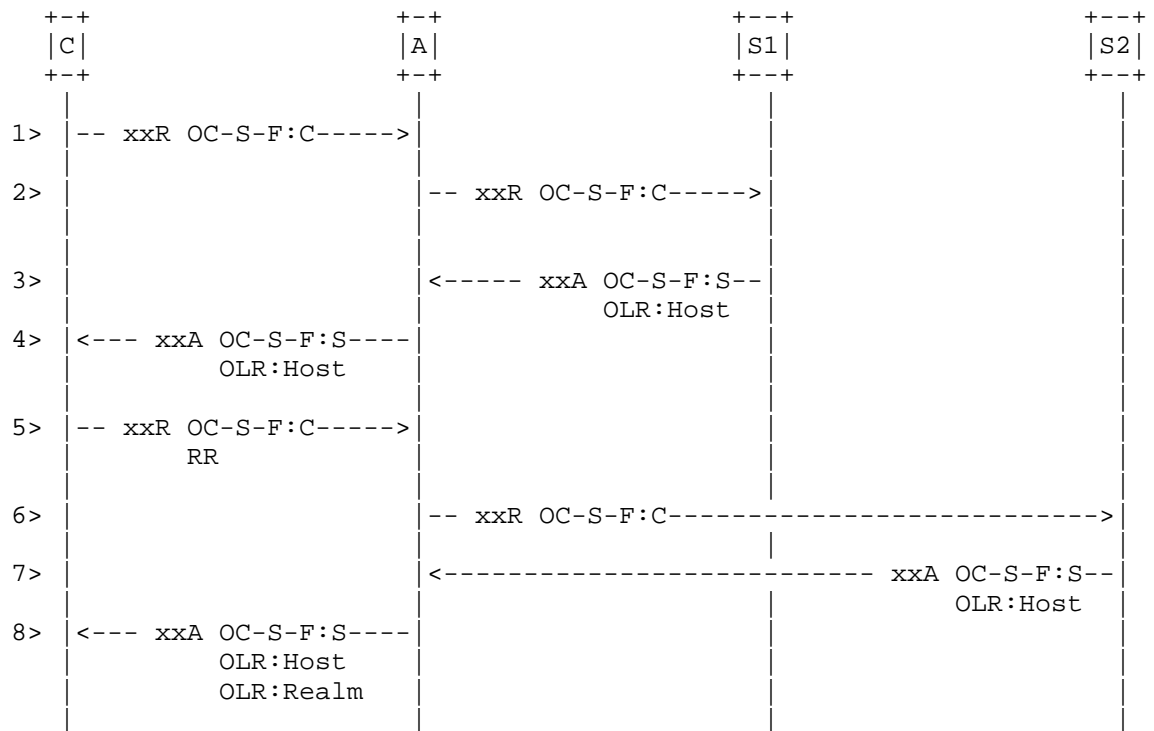


Figure 6

1. Same as in Figure 4.
2. Same as in Figure 4.
3. Same as in Figure 4.
4. Same as in Figure 4.
5. Same as in Figure 4.
6. Same as in Figure 4.
7. Same as in Figure 4, with the addition that S2 also includes an overload report in the answer message.
8. A determines that the available capacity of all servers in the realm has been reduced to the degree that it must generate a realm report. A adds this report to the answer message, in



addition to the existing host report from S2 . C saves overload state associated with the new Realm overload report. For the duration of the realm report the client performs the requested abatement on realm-routed requests.

#### 5.1.3. DOIC Specification Impacts

The following is a list of behavior that needs to be reflected in the DOIC specification.

- o There can be multiple abating nodes for a single overload report. In this use case, A TC handles abatement of host-routed requests. An agent with a direct transport connection to an overloaded node handles abatement of realm-routed requests that would normally be routed to that node.

- o

Note: It is also possible that an agent will handle abatement of host-routed requests, as illustrated in Section 5.3.1.

- o Syntax for the OC-OLR AVP must support multiple OC-OLR AVPs in answer messages.
- o The working group must define a Diameter-Throttled error response that indicates that the request was rejected due to overload and that the request should not be retried.

#### 5.2. Mixed Capabilities

This use case explores the impact of having a different set of DOIC capabilities supported by the TC and one or more agents in the path of the request.

##### 5.2.1. Capability Announcement

Figure 7 illustrates the case. In this figure, "OC-S-F:C" indicates it carries the set of capabilities supported by C. "OC-S-FC:AC" indicates the set of capabilities that A declares to S. "OC-S-FC:S" indicates S's response to the AC set of capabilities. OC-S-FC:AS indicates A's modification to the capabilities selected by S. This is needed in the case where S's capabilities are not compatible with C's.

"OC-S-FC:AC" could indicate the merged capabilities of C and A, based on local policies at A. For example, A could indicate a union or intersection of the its local capabilities with those of C. Alternately, it A could declare an entirely different set of

capabilities towards S. If the capabilities selected between A and S differ from those selected between C and A, A becomes responsible for mapping any overload information it receives from S to fit the capabilities it negotiated with C.

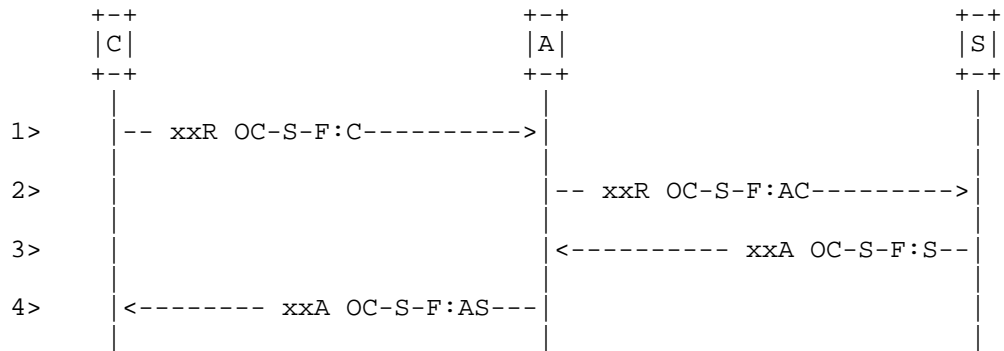


Figure 7

1. C originates a request including OC-S-F:C, indicating the DOIC features supported by C.
2. A inspects OC-S-F:C and determines that A supports features not included. A relays the request, replacing OC-S-F:C APV with an OC-S-F:AC.
3. S responds to the set of advertised features with the OC-S-F:S AVP. There is no change in S's behavior beyond what is specified in [I-D.ietf-dime-ovli] and any other extensions documenting the features in the received OC-S-F AVP.
4. A inspects OC-S-F:S. If necessary A replaces OC-S-F:S with OC-S-F:AS'.

Section 5.2.2 illustrates one example where A needs to OC-S-F:S with OC-S-F:AS'.

#### 5.2.2. Mixed Abatement Algorithms

Figure 8 illustrates one specific type of mixed capabilities. In this case, C only supports the loss abatement algorithm, A supports both loss and rate and S selects rate.

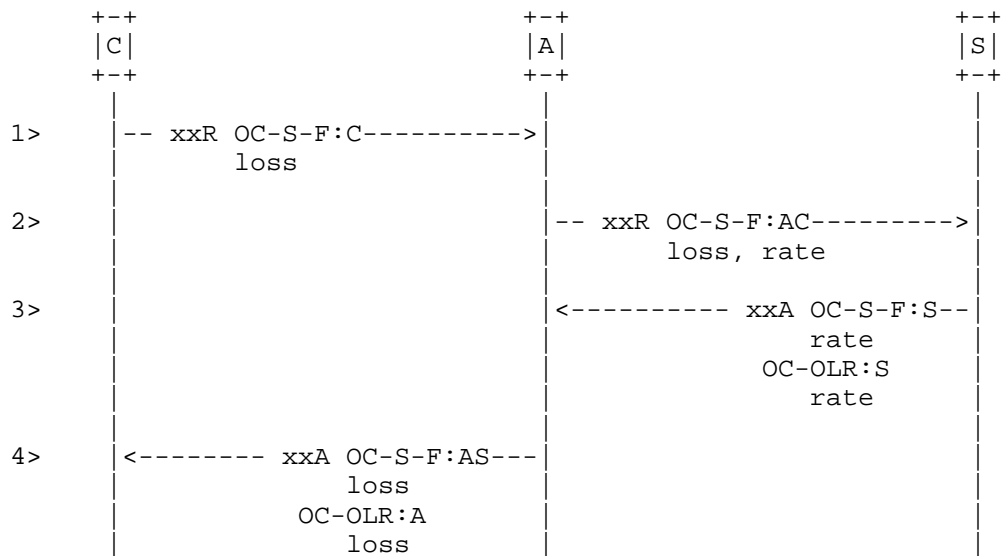


Figure 8

1. C originates a request with OC-S-F:C, indicating support for only the "loss" algorithm.
2. A inspects OC-S-F:C and determines it needs to advertise support for additional capabilities. A removes OC-S-F:C and inserts OC-S-F:AC, indicating support for both the "loss" and "rate" algorithms. A stores the state from OC-S-F:C to be referenced when it receives the associated answer.
3. S responds with OC-S-F:S, indicating that the rate algorithm will be used for overload reports, and OC-OLR:rate, indicating an overload condition with a specific requested rate-limit.
4. A recalls that C did not indicate support the rate algorithm and replaces OC-S-F:S with OC-S-F:AS, which indicates that the loss abatement algorithm will be used for overload reports sent to C. A must enforce the rate limit locally, so it removes OC-OLR:rate. If C's offered load exceeds what A can handle without violating the requested rate-limit, it inserts OC-OLR:A, requesting a traffic reduction using the "loss" algorithm.

This flow assumes that A is able to handle rate-based overload reports, even though the TC cannot. How this is done in practice is implementation specific. In this example, A applies local rate-

limiting, but send loss-based OLRs to C if A cannot handle C's offered load without violating the rate-limit.

If A chose to apply local throttling to enforce the rate limit, instead of sending load-based OLRs back to the TC, the scenario would be closer to that for a TC that did not support DOIC (Section 5.3.1).

#### 5.2.3. DOIC Specification Impacts

An agent must have the ability to replace the OC-S-F AVP in request messages.

An agent must have the ability to remove or replace the OC-S-F AVP in answer messages.

An agent must have the ability to remove or replace the OC-OLR AVP in answer messages.

#### 5.3. Non-Supporting Nodes

This section outlines the impact of agent based scenarios where there is a node that does not support DOIC in the path of a request. There are five variations of this use case:

1. Non-supporting TC.
2. Non-supporting TS.
3. Non-supporting agent between the TC and a DOIC agent.
4. Non-supporting agent between a DOIC agent and the TS.
5. Non-supporting agent between DOIC agents.

##### 5.3.1. Non-Supporting Transaction Client

This section outlines the handling of non-supporting transaction client.

This use case is illustrated in Figure 9. In this case assume that C1 supports DOIC and C2 does not.

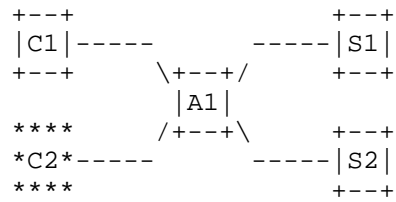


Figure 9

Figure 10 illustrates capability announcement for both the supporting and non-supporting client. This scenario assumes that the capabilities supported by C1 and A are the same.

There is no change from the simple agent use case for transactions originated by C1.

For transactions originated by the non-supporting reacting node C2, A1 determines that C2 does not support DOIC by the absence of an OC-S-F AVP and inserts an OC-S-F AVP indicating the OC features supported by A1.

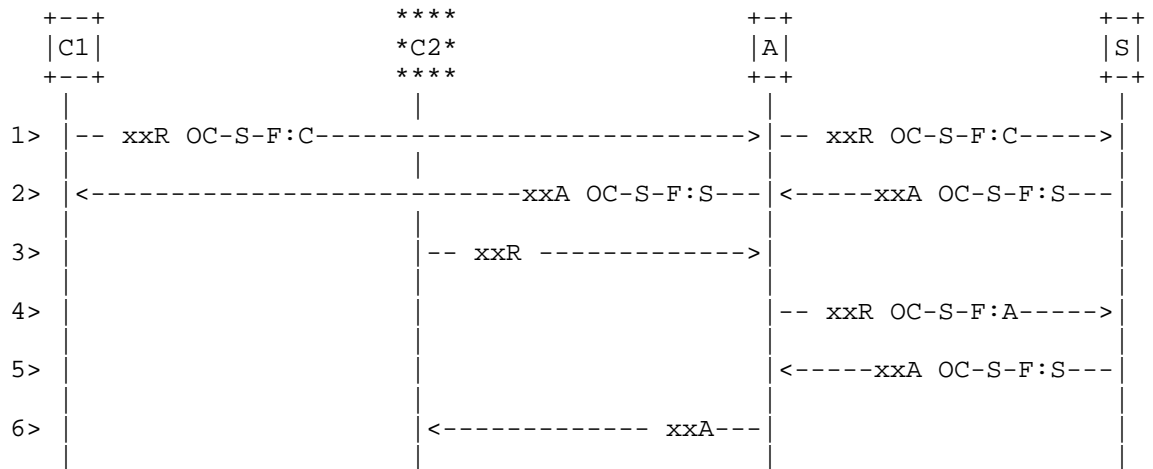


Figure 10

1. C1 supports DOIC and, as such, includes the OC-S-F AVP in all request messages sent. A relays the request to S based on normal request handling.
2. S supports DOIC and, as such, includes the OC-S-F AVP in all response messages sent. A relays the answer to C1 based on normal answer handling.
3. C2 does not support DOIC and, as such, does not insert the OC-S-F AVP into request messages.
4. A recognizes that C2 does not support DOIC, since the request does not contain the OC-S-F AVP. A inserts an OC-S-F AVP that reflects the OC capabilities of A.
5. S does normal DOIC capability announcement handling, inserting the OC-S-F AVP in the answer.
6. A removes the OC-S-F AVP from the answer given that C2 does not support DOIC.

Figure 11 illustrates overload report handling for this scenario.

There is no change in overload handling for requests originated by C1. C1 is responsible for abatement of host routed requests and A is responsible for abatement of realm-routed requests.

For requests originated by C2, it becomes the responsibility of A to handle overload abatement requested by S. In this case A is responsible for abatement of both host-routed and realm-routed requests, as A has a direct transport connection to S. The way an agent handles this in practice is implementation specific. Depending on the algorithm, an agent might be able to treat requests from all non-supporting clients as a pool. More complex implementations might maintain (and certain algorithms might require) the agent to maintain an overload control state machine for each known non-supporting client.

If there were an upstream DOIC agent between A and S then A would no longer have a direct transport connection and would not be able to do abatement of realm-routed requests. It would become the responsibility of the upstream DOIC agent with the transport connection to handle abatement of realm-routed requests.

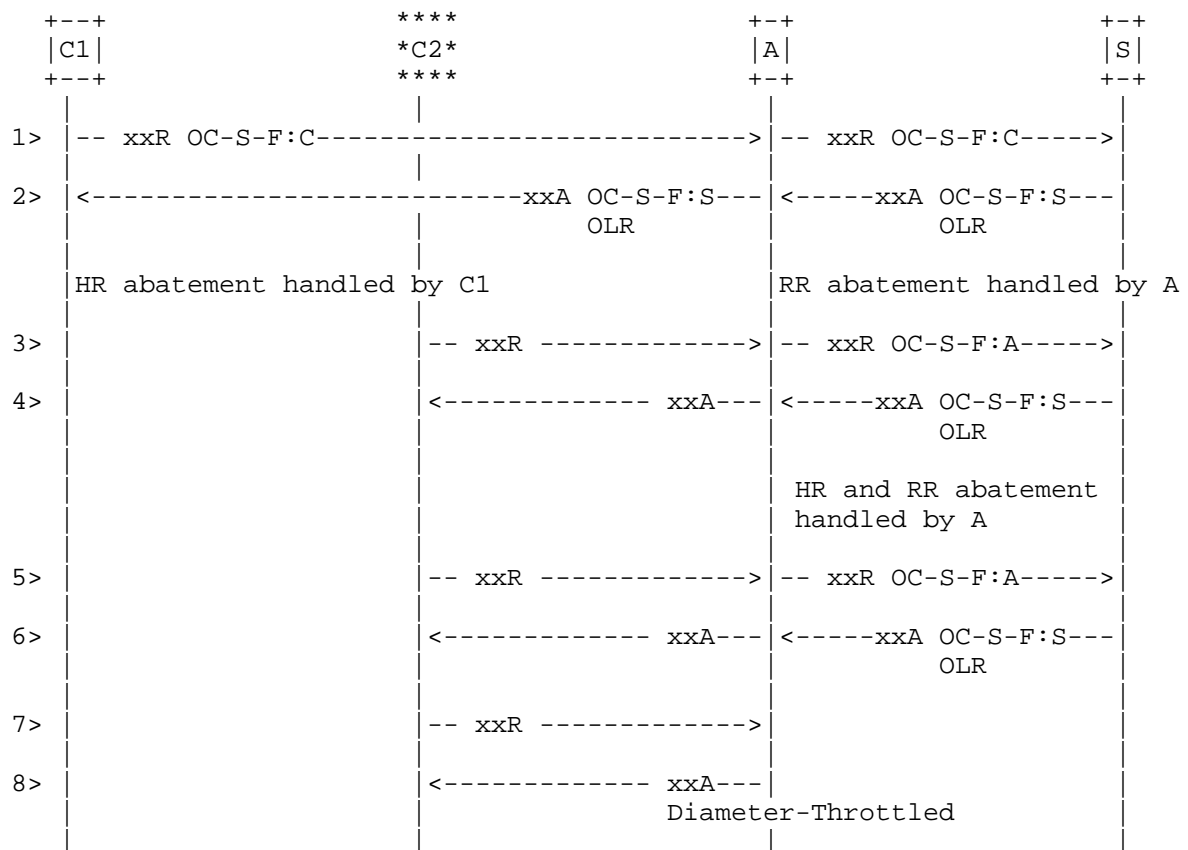


Figure 11

1. Request from C1, a supporting TC.
2. Response indicating S is requesting a reduction in traffic sent due to an overload condition. C1 becomes responsible for abatement of host-routed requests and A becomes responsible for abatement of realm-routed requests.
3. Request from C2, a non-supporting TC. A inserts an OC-S-F AVP.
4. Response indicating that S is overloaded. A stores overload state based on the content of the overload report. A also removed the OC-S-F and OC-OLR AVPs from the answer message. A becomes responsible for abatement handling of all requests originated by C2.

5. Request from non-supporting node originated after the overload report is received. This request survives abatement by A.
6. Response for message that survived abatement by A.
7. Request from non-supporting node originated after the overload report is received. This request does not survive abatement and is rejected by A.
8. Response for request that did not survive abatement by A, with an appropriate error code to indicate the request was throttled and should not be retried.

### 5.3.2. Non-supporting Transaction Server

This section shows the case where there is a mix of transaction servers that support DOIC and those that do not support DOIC.

In this case, it becomes the responsibility of a DOIC agent to become the reporting node for the non-supporting transaction server. The method the agent uses to determine if abatement of traffic is required for the non-supporting node is implementation specific. (For example, an agent may infer that a TS is overloaded by observing Diameter or transport errors, or it may have some proprietary, out-of-band mechanism for learning about TS overload.)

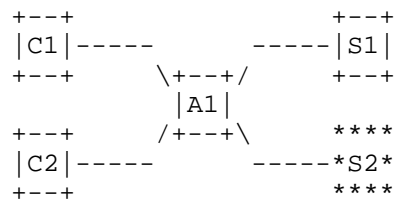


Figure 12



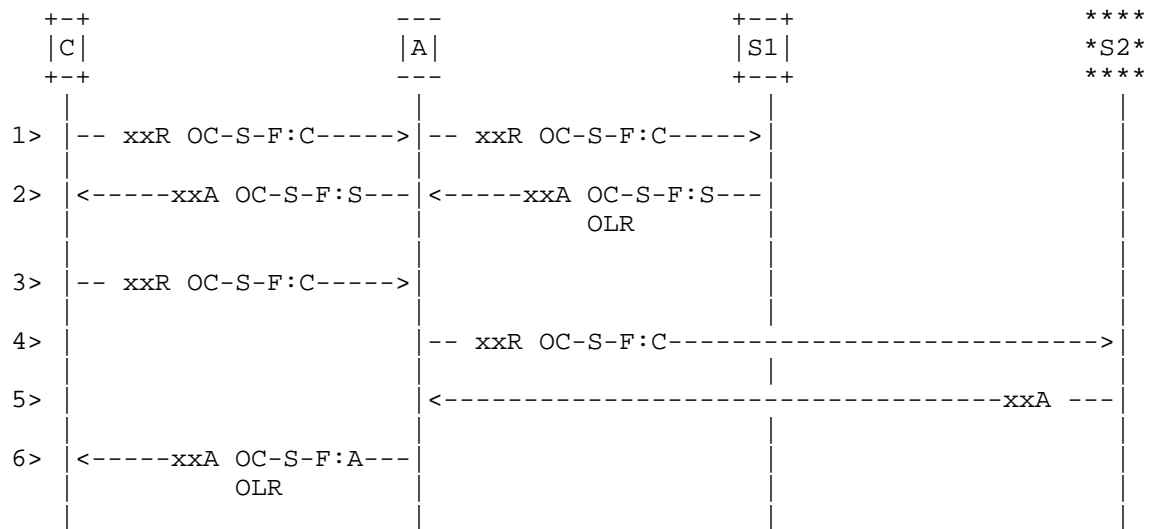


Figure 13

1. Normal DOIC processing resulting in the request being routed to S1.
2. Normal DOIC processing.
3. Normal DOIC processing
4. Normal DOIC processing resulting in the request being routed to S2. The agent doesn't know yet that S2 doesn't support DOIC.
5. S2 does not support DOIC and, as a result, does not insert the OC-S-F AVP in the answer message.
6. A takes on responsibility for becoming the reporting node for S2, and inserts an OC-S-F AVP. In this case A has determined that S2 is in an overload condition and inserts an OC-OLR AVP in the answer message.

C handles the OC-OLR overload report in the same way it handles all OC-OLR reports.

### 5.3.3. Non-Supporting Agent

There are two sub-cases for non-supporting agents.

Figure 14 illustrates the first non-supporting agent case, where the first agent in a chain of agents does not support DOIC.

In this case, A2 picks up the responsibility of handling overload abatement in the case that either C1 or C2 do not support DOIC.

A2 is also responsible for abating realm-routed requests for host reports received from S1 or S2.

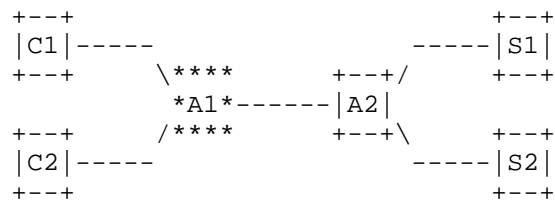


Figure 14

Figure 15 illustrates the second non-supporting agent case, where the last agent in the chain does not support DOIC.

In this scenario, there is no DOIC node that has a direct transport connection with S1 and S2. As a result, there is no DOIC node that can correctly handle abatement of realm-routed requests resulting. In the example, A1 cannot perform diversion, because it cannot control whether any given request goes to S1 or S2. And it cannot correctly determine how much to throttle unless it has advance knowledge of the topology behind A2, which is currently out of scope for DOIC.

As a result, this deployment scenario should be avoided.

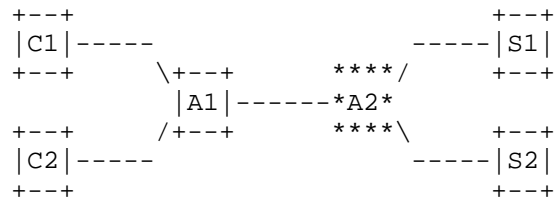


Figure 15

#### 5.3.4. DOIC Specification Impacts

- o Agents must be allowed to insert OC-S-F AVPs into request and answer messages.
- o Agents must be allowed to remove OC-S-F AVPs from request and answer messages.
- o Agents must be able to insert OC-OLR AVPs of type "Host Report" into answer messages. (The ability to insert OC-ORL AVPs of type "Realm Report" is already assumed.)
- o Agents must be allowed to remove OC-OLR AVPs from answer messages.
- o Agents must be allowed to abate host-routed requests.

#### 5.4. Inter Domain Authentication

Figure 16 shows three administrative domains, Dom 1, Dom 2, and Dom 3. Dom 3 does not wish information about the condition of it's network to be shared with Dom 1, but is willing to share overload information with Dom 2 in order to optimize inter-domain traffic.

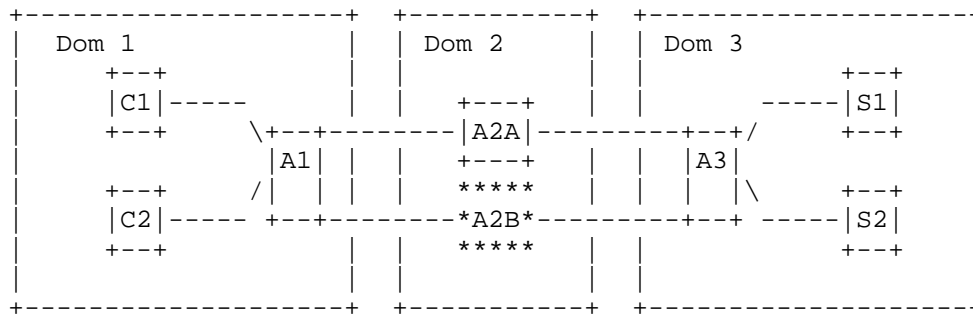


Figure 16

Dom 2 is in the process of incremental deployment of DOIC. Agent A2A supports DOIC, but A2B does not. A1 and A3 are configured so that Diameter requests between them may traverse either A2A or A2B.

Figure 17 shows a Diameter message exchange for each potential route. The originating TC and selected TS are not relevant for the example, so they are omitted from the diagram.

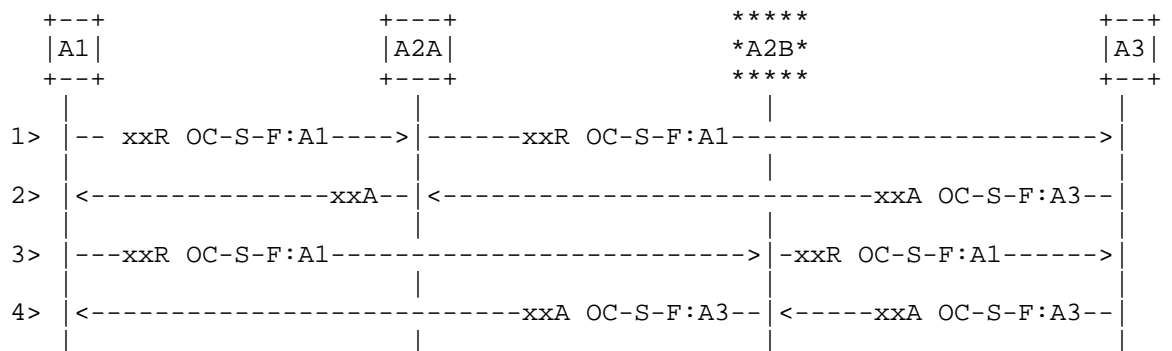


Figure 17

1. A1 sends a request that contains OC-S-F:A1 to A2A. A2A supports the same DOIC capabilities, so it forwards the request with OC-S-F:A1 unchanged.
2. A3 receives the answer from the TC. It forwards the response to A2A, including its capabilities in OC-S-F:A3. A2A is configured to enforce Dom 3's wishes that its overload information not be

sent to Dom 1, so it strips the AVP before forwarding the response back to A1.

3. A1 again sends a request including OC-S-F:A1, but this one traverses A2B. Since A2B does not support DOIC at all, it treats the AVP as an unknown AVP and forwards it unchanged to A3. Note that the OC-S-F AVP observed by A3 is identical to that from step 1.
4. A3 receives the answer from the TC. It mistakenly believes A2B supports DOIC, and therefore forwards the response with it's DOIC capabilities in OC-S-F:A3. Since A2B does not recognize the AVP, it forwards it back to A1 without change.

This is a somewhat contrived example, but it shows a case where Dom 3 leaked information to an untrusted domain, because it could not tell the difference between an OC-S-F AVP received from a trusted peer that supports DOIC, or one forwarded from downstream by a non-supporting peer.

#### 5.4.1. DOIC Specification Impacts

A DOIC supporting node must be able to distinguish between an OC-S-F AVP sent by a peer that supports DOIC, and one sent by a non-adjacent node and forwarded by a non-supporting peer. That is not possible in DOIC at the time of this writing.

The ability to limit overload information to nodes that are authorized to receive it may require the ability to fully attribute a given OC-S-F AVP to the node that included the AVP. Whether this is required, and how an AVP that is forwarded by a DOIC supporting relay is for further study.

#### 6. Recommendations

This section summarizes the recommendations made in previous sections. These recommendations are presented without normative language, but the authors expect that some of the recommendations will require new normative language in [I-D.ietf-dime-ovli]. Others may result in non-normative guidance.

As noted earlier, nothing in this draft should imply that relays are required to deploy DOIC. The majority of these recommendations should be interpreted to allow certain relay behaviors, but not to require them, and to offer architectural guidance on how an operator can best utilize relays in a DOIC deployment if they choose to do so.

## 6.1. General Recommendations

This section describes recommendations that apply to the DOIC mechanism in general:

The working group should define a "Diameter-Throttled" error code, that indicates a request has failed due to overload, and should not be retried.[Section 5.1.3] TCs need to recognize the Diameter-Throttled error code, and interpret it as a final failure for the transaction.

The OC-OLR AVP syntax must allow multiple occurrences in the same Diameter answer message. [Section 5.1.3]

## 6.2. Agent Behavior Recommendations

The discussion in Section 4, Section 3, and Section 5 suggest certain recommendations for DOIC supporting Diameter relay behavior. The authors recommend that language be added to [I-D.ietf-dime-ovli] to the general effect of the following sections:

### 6.2.1. Capabilities Exchange Behaviors

This section describes recommended Agent behaviors with respect to the OC-Supported-Features AVP.

A DOIC supporting agent may act as a reporting-node, a reacting-node, or both.

An agent may act as a reporting node on behalf of a non-supporting TS, an abating node on behalf of a non-supporting TC.[Section 5.3]

An agent that acts as a reacting node must include an OC-Supported-Features in each Diameter request that it forwards in that role. If the inbound request included an OC-Supported-Features AVP, the relay may copy its content to the one in the outbound request, or may replace the contents if it wishes to indicate different DOIC capabilities to upstream nodes. If an inbound request does not contain an OC-Supported-Features AVP, the agent must insert one into the outbound request, indicating the DOIC capabilities of the agent itself.

An agent that acts as a reporting node must include an OC-Supported-Features AVP in each Diameter answer that it forwards in that role. If the agent modified the OC-Supported-Features AVP in the associated request, it must perform a reciprocal modification of the OC-Supported-Features AVP in the response.

An agent that does not support the DOIC mechanism is likely to forward an OC-Supported-Features AVP without modification. A DOIC node must be able to tell between an OC-Supported-Features AVP that was forwarded by such a non-supporting agent, and one inserted or copied by a DOIC-supporting node.[Section 5.4]

#### 6.2.2. Overload Report Behaviors

When a DOIC-supporting relay inserts an OC-Supported-Features AVP (or passes through one received from downstream), it becomes responsible for ensuring that any OLRs it receives from upstream nodes are honored. It can honor an OLR by locally performing overload abatement, delegating abatement to downstream nodes, or a combination of both.

If a relay can honor the OLR by locally diverting traffic, it should do so before resorting to throttling. For example, if a relay receives a realm report from its upstream peer, and has other less-overloaded peers that are valid for the realm and application, it diverts traffic to the less overloaded peers as needed. The relay should apply any knowledge it has of the peers' relative load and capacity in determining how to divert traffic. Note that only a relay that has a direct peer relationship with the servers in question can effectively perform diversion, since a Diameter node cannot directly control how upstream relays will route requests.[Section 5.1]

When an overload condition requires throttling of traffic, an agent should delegate that throttling to downstream nodes if at all possible. Depending on local policy and the nature of the overload condition, this means the agent either originates a new OLR to send downstream, or forwards an OLR received from upstream. For example, if a relay receives a host report (which usually requires traffic throttling), the relay typically forwards that report downstream. The relay may modify the report based on local policy.

If an agent needs to perform local throttling, it must explicitly reject each throttled request with a "Diameter-Throttled" error code.[Section 5.1]

There may be circumstances where an agent must perform local throttling. An obvious example is when downstream nodes do not support DOIC, that is, requests from downstream nodes do not contain OC-Supported-Features AVPs. Mismatched upstream and downstream capabilities may require local throttling. For example, if a relay uses a rate-limiting abatement algorithm upstream, but downstream devices do not support rate-limiting, it may have to locally throttle traffic to meet its upstream

abatement commitment. It might still invoke the "loss" algorithm downstream in order to reduce the amount of traffic that must be locally throttled.[Section 5.2, Section 5.3]

A relay should apply all the information at hand to determine upstream overload. For example, if a relay receives a host-report from a directly attached TS, that relay can reasonably infer that the overload condition applies to all traffic for the realm, and perform local abatement by diverting realm-routed traffic to other servers. If there is insufficient capacity to do so, then it can generate realm-reports downstream. A relay might also have knowledge of the overload or load state of other nodes through some non-DOIC mechanism.[Section 5.1]

Finally, a relay should not generate or forward OLRs in a way likely to cause redundant abatement. For example, if a relay locally throttles traffic due to a "loss" algorithm OLR, it should not forward the OLR downstream where other nodes will also apply abatement to the same traffic. [Section 5.3]

The idea of redundant abatement is at least somewhat specific to the algorithm. For example, a rate-limiting algorithm might allow both local and delegated abatement, since the algorithm creates a maximum rate limit. On the other hand, the "loss" algorithm requests a percentage reduction. If a relay receives an OLR for a 10 percentage reduction, applies local throttling, and also forwards the OLR downstream, the 10% reduction may be applied twice.

## 7. Security Considerations

Several use cases in this document involve Agents inserting, removing, or modifying DOIC related AVPs. [RFC6733] does not allow "relay" agents to modify any part of a Diameter message except for routing information. This has one of two implications; either relay agents cannot take an active role in DOIC, or the DOIC AVPs should be designated as "routing AVPs". The authors recommend the second.

But regardless of whether a relay is allowed to modify DOIC AVPs, proxy agents will almost certainly need to do so. Diameter currently only offers hop-by-hop integrity protection of message contents, but the DIME working group is considering requirements for end-to-end protection [I-D.ietf-dime-e2e-sec-req] at the time of this writing. Those requirements currently recognize that different AVPs may require different security treatments. The working group should carefully consider how DOIC will interact with end-to-end security when it is completed.



Until such end-to-end protection is deployed, Diameter follows a fundamentally transitive trust model. Adjacent nodes can authenticate each other's identity, and protect exchanged messages from tampering or eavesdropping. But Diameter nodes have no way of authenticating message content received from non-adjacent nodes, other than trusting the immediate peer to do the right thing.

DOIC related information may be sensitive, and could be destructive if forged or modified by unauthorized parties. DOIC nodes must trust DOIC supporting peers to ensure that no unauthorized parties insert overload reports, and to ensure that reports are not delivered to unauthorized parties. Peers that do not support DOIC cannot be expected to enforce such policies.

At the time of this writing, DOIC provides no way for a supporting node to distinguish between a DOIC AVP from a immediate peer that supports DOIC, and one forwarded by a non-supporting peer. This issue needs to be addressed before DOIC can meet requirements 27, 28, and 29 of [RFC7068]

## 8. References

### 8.1. Normative References

[DOIC-rate]

Donovan, S. and E. Noel, "Diameter Agent Overload", February 2014.

[I-D.ietf-dime-ovli]

Korhonen, J., Donovan, S., Campbell, B., and L. Morand, "Diameter Overload Indication Conveyance", draft-ietf-dime-ovli-02 (work in progress), March 2014.

[RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J. Loughney, "Diameter Credit-Control Application", RFC 4006, August 2005.

[RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

[agent-overload]

Donovan, S., "Diameter Agent Overload", March 2014.

### 8.2. Informative References

- [I-D.ietf-dime-e2e-sec-req]  
Tschofenig, H., Korhonen, J., Zorn, G., and K. Pillay,  
"Diameter AVP Level Security: Scenarios and Requirements",  
draft-ietf-dime-e2e-sec-req-01 (work in progress), October  
2013.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629,  
June 1999.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC  
Text on Security Considerations", BCP 72, RFC 3552, July  
2003.
- [RFC7068] McMurtry, E. and B. Campbell, "Diameter Overload Control  
Requirements", RFC 7068, November 2013.

## Authors' Addresses

Steve Donovan  
Oracle  
Frisco, Texas  
USA

Phone: +1  
Email: [srdonovan@usdonovans.com](mailto:srdonovan@usdonovans.com)

Ben Campbell  
Oracle  
Frisco, Texas  
USA

Phone: +1  
Email: [ben@nostrum.com](mailto:ben@nostrum.com)

Diameter Maintenance and Extensions (DIME)  
Internet-Draft  
Intended status: Informational  
Expires: June 22, 2014

L. Morand, Ed.  
Orange Labs  
V. Fajardo

H. Tschofenig  
Nokia Siemens Networks  
December 19, 2013

Diameter Applications Design Guidelines  
draft-ietf-dime-app-design-guide-21

Abstract

The Diameter base protocol provides facilities for protocol extensibility enabling to define new Diameter applications or modify existing applications. This document is a companion document to the Diameter Base protocol that further explains and clarifies the rules to extend Diameter. It is meant as a guidelines document and therefore as informative in nature.

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 June 22, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	3
2. Terminology . . . . .	3
3. Overview . . . . .	3
4. Reusing Existing Diameter Applications . . . . .	5
4.1. Adding a New Command . . . . .	5
4.2. Deleting an Existing Command . . . . .	6
4.3. Reusing Existing Commands . . . . .	6
4.3.1. Adding AVPs to a Command . . . . .	7
4.3.2. Deleting AVPs from a Command . . . . .	8
4.4. Reusing Existing AVPs . . . . .	9
4.4.1. Setting of the AVP Flags . . . . .	9
4.4.2. Reuse of AVP of Type Enumerated . . . . .	9
5. Defining New Diameter Applications . . . . .	10
5.1. Introduction . . . . .	10
5.2. Defining New Commands . . . . .	10
5.3. Use of Application-Id in a Message . . . . .	11
5.4. Application-Specific Session State Machines . . . . .	11
5.5. Session-Id AVP and Session Management . . . . .	12
5.6. Use of Enumerated Type AVPs . . . . .	12
5.7. Application-Specific Message Routing . . . . .	14
5.8. Translation Agents . . . . .	15
5.9. End-to-End Application Capabilities Exchange . . . . .	15
5.10. Diameter Accounting Support . . . . .	16
5.11. Diameter Security Mechanisms . . . . .	18
6. Defining Generic Diameter Extensions . . . . .	18
7. Guidelines for Registrations of Diameter Values . . . . .	19
8. IANA Considerations . . . . .	21
9. Security Considerations . . . . .	21
10. Contributors . . . . .	22

11. Acknowledgments . . . . .	22
12. Informative References . . . . .	22
Authors' Addresses . . . . .	24

## 1. Introduction

The Diameter base protocol provides facilities to extend Diameter (see Section 1.3 of [RFC6733]) to support new functionality. In the context of this document, extending Diameter means one of the following:

1. Addition of new functionality to an existing Diameter application without defining a new application.
2. Addition of new functionality to an existing Diameter application that requires the definition of a new application.
3. The definition of an entirely new Diameter application to offer functionality not supported by existing applications.
4. The definition of a new generic functionality that can be reused across different applications.

All of these choices are design decisions that can be done by any combination of reusing existing or defining new commands, AVPs or AVP values. However, application designers do not have complete freedom when making their design. A number of rules have been defined in [RFC6733] that place constraints on when an extension requires the allocation of a new Diameter application identifier or a new command code value. The objective of this document is the following:

- o Clarify the Diameter extensibility rules as defined in the Diameter base protocol.
- o Discuss design choices and provide guidelines when defining new applications.
- o Present trade-off choices.

## 2. Terminology

This document reuses the terminology defined in [RFC6733].

## 3. Overview

As designed, the Diameter base protocol [RFC6733] can be seen as a two-layer protocol. The lower layer is mainly responsible for

managing connections between neighboring peers and for message routing. The upper layer is where the Diameter applications reside. This model is in line with a Diameter node having an application layer and a peer-to-peer delivery layer. The Diameter base protocol document defines the architecture and behavior of the message delivery layer and then provides the framework for designing Diameter applications on the application layer. This framework includes definitions of application sessions and accounting support (see Section 8 and Section 9 of [RFC6733]). Accordingly, a Diameter node is seen in this document as a single instance of a Diameter message delivery layer and one or more Diameter applications using it.

The Diameter base protocol is designed to be extensible and the principles are described in the Section 1.3 of [RFC6733]. As a summary, Diameter can be extended by:

1. Defining new AVP values
2. Creating new AVPs
3. Creating new commands
4. Creating new applications

As a main guiding principle, the recommendation is: "try to re-use as much as possible!". It will reduce the time to finalize specification writing, and it will lead to a smaller implementation effort as well as reduce the need for testing. In general, it is clever to avoid duplicate effort when possible.

However, re-use is not appropriate when the existing functionality does not fit the new requirement and/or the re-use leads to ambiguity.

The impact on extending existing applications can be categorized into two groups:

Minor Extension: Enhancing the functional scope of an existing application by the addition of optional features to support. Such enhancement has no backward compatibility issue with the existing application.

A typical example would be the definition of a new optional AVP for use in an existing command. Diameter implementations supporting the existing application but not the new AVP will simply ignore it, without consequences for the Diameter message handling. The standardization effort will be fairly small.

Major Extension:    Enhancing an application that requires the definition of a new Diameter application.

Typical examples would be the creation of a new command for providing functionality not supported by existing applications or the definition of a new AVP with the M-bit set to be carried in an existing command. For such extension, a significant specification effort is required and a careful approach is recommended.

We would also like to remind that the definition of a new Diameter application and the definition of a new command should be something to avoid as much as possible. In the past, there has been some reluctance to define new commands and new applications. With the modified extensibility rules provided by [RFC6733], registering new commands and new applications does not lead to additional overhead for the specification author in terms of standardization process. Registering new functionality (new commands, new AVPs, new applications, etc.) with IANA remains important to avoid namespace collisions, which will likely lead to deployment problems.

#### 4.    Reusing Existing Diameter Applications

An existing application may need to be enhanced to fulfill new requirements and these modifications can be at the command level and/or at the AVP level. The following sections describe the possible modifications that can be performed on existing applications and their related impact.

##### 4.1.    Adding a New Command

Adding a new command is considered as a major extension and requires a new Diameter application to be defined. Adding a new command to an application means either defining a completely new command or importing the command's Command Code Format (CCF) syntax from another application whereby the new application inherits some or all of the functionality of the application where the command came from. In the former case, the decision to create a new application is straightforward since this is typically a result of adding a new functionality that does not exist yet. For the latter, the decision to create a new application will depend on whether importing the command in a new application is more suitable than simply using the existing application as it is in conjunction with any other application. Therefore, a case by case study of each application requirement should be applied.

An example considers the Diameter EAP application [RFC4072] and the Diameter NASREQ application [RFC4005]. When network access authentication using EAP is required, the Diameter EAP commands

(Diameter-EAP-Request/Diameter-EAP-Answer) are used; otherwise the NASREQ application will be used. When the Diameter EAP application is used, the accounting exchanges defined in Diameter NASREQ may be used.

However, in general, it is difficult to come to a hard guideline, and so a case-by-case study of each application requirement should be applied. Before adding or importing a command, application designers should consider the following:

- o Can the new functionality be fulfilled by creating a new command independent from any existing command? In this case, the resulting new application and the existing application can work independent of, but cooperating with each other.
- o Can the existing command be reused without major extensions and therefore without the need for the definition of a new application, e.g., new functionality introduced by the creation of new optional AVPs.

Note: Importing commands too liberally could result in a monolithic and hard to manage application supporting too many different features.

#### 4.2. Deleting an Existing Command

Although this process is not typical, removing a command from an application requires a new Diameter application to be defined. This is due to the fact that the reception of the deleted command would systematically result in a protocol error (i.e., `DIAMETER_COMMAND_UNSUPPORTED`).

It is unusual to delete an existing command from an application for the sake of deleting it or the functionality it represents. This normally indicates of a flawed design. An exception might be if the intent of the deletion is to create a newer version of the same application that is somehow simpler than the previous version.

#### 4.3. Reusing Existing Commands

This section discusses rules in adding and/or deleting AVPs from an existing command of an existing application. The cases described in this section may not necessarily result in the creation of new applications.

From a historical point of view, it is worth to note that there was a strong recommendation to re-use existing commands in the [RFC3588] to



prevent rapid depletion of code values available for vendor-specific commands. However, [RFC6733] has relaxed the allocation policy and enlarged the range of available code values for vendor-specific applications. Although reuse of existing commands is still recommended, protocol designers can consider defining a new command when it provides a solution more suitable than the twisting of an existing command's use and applications.

#### 4.3.1. Adding AVPs to a Command

Based on the rules in [RFC6733], AVPs that are added to an existing command can be categorized into:

- o Mandatory (to understand) AVPs. As defined in [RFC6733], these are AVPs with the M-bit flag set in this command, which means that a Diameter node receiving them is required to understand not only their values but also their semantics. Failure to do so will cause an message handling error. This is regardless of whether these AVPs are required or optional as specified by the command's Command Code Format (CCF) syntax .
- o Optional (to understand) AVPs. As defined in [RFC6733], these are AVPs with the M-bit flag cleared in this command. A Diameter node receiving these AVPs can simply ignore them if it does not support them.

NOTE: As stated in RFC6733, the M-bit setting for a given AVP is relevant to an application and each command within that application that includes the AVP.

The rules are strict in the case where the AVPs to be added in an exiting command are mandatory to understand, i.e., they have the M-bit set. A mandatory AVP cannot be added to an existing command without defining a new Diameter application, as stated in [RFC6733]. This falls into the "Major Extensions" category. Despite the clarity of the rule, ambiguity still arises when evaluating whether a new AVP being added should be mandatory to begin with. Application designers should consider the following questions when deciding about the M-bit for a new AVP:

- o Would it be required for the receiving side to be able to process and understand the AVP and its content?
- o Would the new AVPs change the state machine of the application?
- o Would the presence of the new AVP lead to a different number of round-trips, effectively changing the state machine of the application?

- o Would the new AVP be used to differentiate between old and new versions of the same application whereby the two versions are not backward compatible?
- o Would the new AVP have duality in meaning, i.e., be used to carry application-related information as well as to indicate that the message is for a new application?

If the answer to at least one of the questions is "yes" then the M-bit has to be set for the new AVP. This list of questions is non-exhaustive and other criteria can be taken into account in the decision process.

If application designers are instead contemplating the use of optional AVPs, i.e., with the M-bit cleared, then the following are some of the pitfalls that should be avoided:

- o Use of optional AVPs with intersecting meaning. One AVP has partially the same usage and meaning as another AVP. The presence of both can lead to confusion.
- o An optional AVPs with dual purpose, i.e., to carry application data as well as to indicate support for one or more features. This has a tendency to introduce interpretation issues.
- o Adding one or more optional AVPs and indicating (usually within descriptive text for the command) that at least one of them has to be present in the command. This essentially circumventing the ABNF and is equivalent to adding a mandatory AVP to the command.

These practices generally result in interoperability issues and should be avoided as much as possible.

#### 4.3.2. Deleting AVPs from a Command

Application designers may want to reuse an existing command but some of the AVP present in the command's CCF syntax specification may be irrelevant for the functionality foreseen to be supported by this command. It may be then tempting to delete those AVPs from the command.

The impacts of deleting an AVP from a command depends on its command code format specification and M-bit setting:

- o Deleting an AVP that is indicated as { AVP } in the command's CCF syntax specification (regardless of the M-bit setting).

In this case, a new command code and subsequently a new Diameter application have to be specified.

- o Deleting an AVP, which has the M-bit set, and is indicated as [ AVP ] in the command's CCF syntax specification.

No new command code has to be specified but the definition of a new Diameter application is required.

- o Deleting an AVP, which has the M-bit cleared, and is indicated as [ AVP ] in the command's CCF syntax specification.

In this case, the AVP can be deleted without consequences.

If possible, application designers should attempt to reuse the command's CCF syntax specification without modification and simply ignore (but not delete) any optional AVP that will not be used. This is to maintain compatibility with existing applications that will not know about the new functionality as well as maintain the integrity of existing dictionaries.

#### 4.4. Reusing Existing AVPs

This section discusses rules in reusing existing AVP when reusing an existing command or defining a new command in a new application.

##### 4.4.1. Setting of the AVP Flags

When reusing AVPs in a new application, the AVP flag setting, such as the mandatory flag ('M'-bit), has to be re-evaluated for a new Diameter application and, if necessary, even for every command within the application. In general, for AVPs defined outside of the Diameter base protocol, the characteristics of an AVP are tied to its role within an application and the commands.

All other AVP flags shall remain unchanged.

##### 4.4.2. Reuse of AVP of Type Enumerated

When reusing an AVP of type Enumerated in a command for a new application, it is recommended to avoid modifying the set of valid values defined for this AVP. Modifying the set of Enumerated values includes adding a value or deprecating the use of a value defined initially for the AVP. Modifying the set of values will impact the application defining this AVP and all the applications using this AVP with potential interoperability issues. When the full range of values defined for this Enumerated AVP is not suitable for the new application, it is recommended to define a new AVP to avoid backwards compatibility issues with existing implementations.

## 5. Defining New Diameter Applications

### 5.1. Introduction

This section discusses the case where new applications have requirements that cannot be fulfilled by existing applications and would require definition of completely new commands, AVPs and/or AVP values. Typically, there is little ambiguity about the decision to create these types of applications. Some examples are the interfaces defined for the IP Multimedia Subsystem of 3GPP, e.g., Cx/Dx ([TS29.228] and [TS29.229]), Sh ([TS29.328] and [TS29.329]) etc.

Application designers should try to import existing AVPs and AVP values for any newly defined commands. In certain cases where accounting will be used, the models described in Section 5.10 should also be considered.

Additional considerations are described in the following sections.

### 5.2. Defining New Commands

As a general recommendation, commands should not be defined from scratch. It is instead recommended to re-use an existing command offering similar functionality and use it as a starting point.

Moreover, the new command's CCF syntax specification should be carefully defined when considering applicability and extensibility of the application. If most of the AVPs contained in the command are indicated as fixed or required, it might be difficult to reuse the same command and therefore the same application in a slightly changed environment. Defining a command with most of the AVPs indicated as optional must not be seen as a sub-optimal design introducing too much flexibility in the protocol. The protocol designers are only advised to clearly state the condition of presence of these AVPs and properly define the corresponding behaviour of the Diameter nodes when these AVPs are absent from the command.

Note: As a hint for protocol designers, it is not sufficient to just look at the command's CCF syntax specification. It is also necessary to carefully read through the accompanying text in the specification.

In the same way, the CCF syntax specification should be defined such that it will be possible to add any arbitrary optional AVPs with the M-bit cleared (including vendor-specific AVPs) without modifying the application. For this purpose, it is strongly recommended to add "[AVP]" in the command's CCF, which allows the addition of any arbitrary AVP as described in [RFC6733].

### 5.3. Use of Application-Id in a Message

When designing new applications, designers should specify that the Application Id carried in all session-level messages must be the Application Id of the application using those messages. This includes the session-level messages defined in Diameter base protocol, i.e., RAR/RAA, STR/STA, ASR/ASA and possibly ACR/ACA in the coupled accounting model, see Section 5.10. Some existing specifications do not adhere to this rule for historical reasons. However, this guidance should be followed to avoid routing problems.

In general, when a new application has been allocated with a new Application Id and it also reuses existing commands with or without modifications, it must use the newly allocated Application Id in the header and in all relevant Application Id AVPs (Auth-Application-Id or Acct-Application-Id) present in the commands message body.

Additionally, application designs using Vendor-Specific-Application-Id AVP should not use the Vendor-Id AVP to further dissect or differentiate the vendor-specification Application Id. Diameter routing is not based on the Vendor-Id. As such, the Vendor-Id should not be used as an additional input for routing or delivery of messages. The Vendor-Id AVP is an informational AVP only and kept for backward compatibility reasons.

### 5.4. Application-Specific Session State Machines

Section 8 of [RFC6733] provides session state machines for authentication, authorization and accounting (AAA) services and these session state machines are not intended to cover behavior outside of AAA. If a new application cannot clearly be categorized into any of these AAA services, it is recommended that the application defines its own session state machine. Support for server-initiated request is a clear example where an application-specific session state machine would be needed, for example, the Rw interface for ITU-T push model (cf.[Q.3303.3]).

### 5.5. Session-Id AVP and Session Management

Diameter applications are usually designed with the aim of managing user sessions (e.g., Diameter network access session (NASREQ) application [RFC4005]) or specific service access session (e.g., Diameter SIP application [RFC4740]). In the Diameter base protocol, session state is referenced using the Session-Id AVP. All Diameter messages that use the same Session-Id will be bound to the same session. Diameter-based session management also implies that both Diameter client and server (and potentially proxy agents along the path) maintain session state information.

However, some applications may not need to rely on the Session-Id to identify and manage sessions because other information can be used instead to correlate Diameter messages. Indeed, the User-Name AVP or any other specific AVP can be present in every Diameter message and used therefore for message correlation. Some applications might not require the notion of Diameter session concept at all. For such applications, the Auth-Session-State AVP is usually set to `NO_STATE_MAINTAINED` in all Diameter messages and these applications are therefore designed as a set of stand-alone transactions. Even if an explicit access session termination is required, application-specific commands are defined and used instead of the Session-Termination-Request/Answer (STR/STA) or Abort-Session-Request/Answer (ASR/ASA) defined in the Diameter base protocol. In such a case, the Session-Id is not significant.

Based on these considerations, protocol designers should carefully appraise whether the application currently defined relies on its own session management concept or whether the Session-Id defined in the Diameter base protocol would be used for correlation of messages related to the same session. If not, the protocol designers could decide to define application commands without the Session-Id AVP. If any session management concept is supported by the application, the application documentation must clearly specify how the session is handled between client and server (as possibly Diameter agents in the path).

### 5.6. Use of Enumerated Type AVPs

The type Enumerated was initially defined to provide a list of valid values for an AVP with their respective interpretation described in the specification. For instance, AVPs of type Enumerated can be used to provide further information on the reason for the termination of a session or a specific action to perform upon the reception of the request.

As described in the section 4.4.2 above, defining an AVP of type Enumerated presents some limitations in term of extensibility and reusability. Indeed, the finite set of valid values defined at the definition of the AVP of type Enumerated cannot be modified in practice without causing backward compatibility issues with existing implementations. As a consequence, AVPs of Type Enumerated cannot be extended by adding new values to support new capabilities. Diameter protocol designers are then strongly advised to carefully consider before defining an Enumerated AVP whether the set of values will remain unchanged or new values may be required in a near future. If such extension is foreseen or cannot be avoided, it is recommended to rather define AVPs of type Unsigned32 or Unsigned64 in which the data field would contain an address space representing "values" that would have the same use of Enumerated values.

For instance, an AVP describing possible access networks would be defined as follow:

Access-Network-Type AVP (XXX) is of type Unsigned32 and contains an 32-bit address space representing types of access networks. This application defines the following classes of access networks, all identified by the thousands digit in the decimal notation:

- o 1xxx (Mobile Access Networks)
- o 2xxx (Fixed Access Network)
- o 3xxx (Wireless Access Networks)

Values that fall within the Mobile Access Networks category are used to inform a peer that a request has been sent for a user attached to a mobile access networks. The following values are defined in this application:

1001: 3GPP-GERAN

TBD.

1002: 3GPP-UTRAN-FDD

TBD.

Unlike Enumerated AVP, any new value can be added in the address space defined by this Unsigned32 AVP without modifying the definition of the AVP. There is therefore no risk of backward compatibility issue, especially when intermediate nodes may be present between Diameter endpoints.

In the same line, AVPs of type Enumerated are too often used as a simple Boolean flag, indicating for instance a specific permission or capability, and therefore only two values are defined, e.g., TRUE/FALSE, AUTHORIZED/UNAUTHORIZED or SUPPORTED/UNSUPPORTED. This is a sub-optimal design since it limits the extensibility of the application: any new capability/permission would have to be supported by a new AVP or new Enumerated value of the already defined AVP, with the backward compatibility issues described above. Instead of using an Enumerated AVP for a Boolean flag, protocol designers are again encouraged to use AVPs of type Unsigned32 or Unsigned64 AVP in which the data field would be defined as bit mask whose bit settings are described in the relevant Diameter application specification. Such AVPs can be reused and extended without major impact on the Diameter application. The bit mask should leave room for future additions. Examples of AVPs that use bit masks are the Session-Binding AVP defined in [RFC6733] and the MIP6-Feature-Vector AVP defined in [RFC5447].

#### 5.7. Application-Specific Message Routing

As described in [RFC6733], a Diameter request that needs to be sent to a home server serving a specific realm, but not to a specific server (such as the first request of a series of round trips), will contain a Destination-Realm AVP and no Destination-Host AVP.

For such a request, the message routing usually relies only on the Destination- Realm AVP and the Application Id present in the request message header. However, some applications may need to rely on the User-Name AVP or any other application-specific AVP present in the request to determine the final destination of a request, e.g., to find the target AAA server hosting the authorization information for a given user when multiple AAA servers are addressable in the realm.

In such a context, basic routing mechanisms described in [RFC6733] are not fully suitable, and additional application-level routing mechanisms have to be described in the application documentation to provide such specific AVP-based routing. Such functionality will be basically hosted by an application-specific proxy agent that will be responsible for routing decisions based on the received specific AVPs.

Examples of such application-specific routing functions can be found in the Cx/Dx applications ([TS29.228] and [TS29.229]) of the 3GPP IP Multimedia Subsystem, in which the proxy agent (Subscriber Location Function aka SLF) uses specific application-level identities found in the request to determine the final destination of the message.



Whatever the criteria used to establish the routing path of the request, the routing of the answer has to follow the reverse path of the request, as described in [RFC6733], with the answer being sent to the source of the received request, using transaction states and hop-by-hop identifier matching. In particular, this ensures that the Diameter Relay or Proxy agents in the request routing path will be able to release the transaction state upon receipt of the corresponding answer, avoiding unnecessary failover. Application designers are strongly dissuaded from modifying the answer-routing principles described in [RFC6733] when defining a new application.

#### 5.8. Translation Agents

As defined in [RFC6733], a translation agent is a device that provides interworking between Diameter and another protocol (e.g., RADIUS).

In the case of RADIUS, it was initially thought that defining the translation function would be straightforward by adopting few basic principles, e.g., by the use of a shared range of code values for RADIUS attributes and Diameter AVPs. Guidelines for implementing a RADIUS-Diameter translation agent were put into RFC 4005 ([RFC4005]).

However, it was acknowledged that such translation mechanism was not so obvious and deeper protocol analysis was required to ensure efficient interworking between RADIUS and Diameter. Moreover, the interworking requirements depend on the functionalities provided by the Diameter application under specification, and a case-by-case analysis will be required.

Therefore, protocol designers cannot assume the availability of a "standard" Diameter-to-RADIUS gateways agent when planning to interoperate with the RADIUS infrastructure. They should specify the required translation mechanism along with the Diameter application, if needed. This recommendation applies for any kind of translation.

#### 5.9. End-to-End Application Capabilities Exchange

New Diameter applications can rely on optional AVPs to exchange application-specific capabilities and features. These AVPs can be exchanged on an end-to-end basis at the application layer. Examples of this can be found with the MIP6-Feature-Vector AVP in [RFC5447] and the QoS-Capability AVP in [RFC5777].

The end-to-end capabilities AVPs formalize the addition of new optional functionality to existing applications by announcing support for it. Applications that do not understand these AVPs can discard them upon receipt. Receivers of these AVPs can discover the

additional functionality supported by the end-point originating the request and behave accordingly when processing the request. Senders of these AVPs can safely assume the receiving end-point does not support any functionality carried by the AVP if it is not present in corresponding response. This is useful in cases where deployment choices are offered, and the generic design can be made available for a number of applications.

When used in a new application, protocol designers should clearly specify this end-to-end capabilities exchange and the corresponding behaviour of the Diameter nodes supporting the application.

It is also important to note that this end-to-end capabilities exchange relies on the use of optional AVPs is not meant as a generic mechanism to support extensibility of Diameter applications with arbitrary functionality. When the added features drastically change the Diameter application or when Diameter agents have to be upgraded to support the new features, a new application should be defined.

#### 5.10. Diameter Accounting Support

Accounting can be treated as an auxiliary application that is used in support of other applications. In most cases, accounting support is required when defining new applications. This document provides two possible models for using accounting:

##### Split Accounting Model:

In this model, the accounting messages will use the Diameter base accounting Application Id (value of 3). The design implication for this is that the accounting is treated as an independent application, especially for Diameter routing. This means that accounting commands emanating from an application may be routed separately from the rest of the other application messages. This may also imply that the messages end up in a central accounting server. A split accounting model is a good design choice when:

- \* The application itself does not define its own accounting commands.
- \* The overall system architecture permits the use of centralized accounting for one or more Diameter applications.

Centralizing accounting may have advantages but there are also drawbacks. The model assumes that the accounting server can differentiate received accounting messages. Since the received accounting messages can be for any application and/or service, the

accounting server has to have a method to match accounting messages with applications and/or services being accounted for. This may mean defining new AVPs, checking the presence, absence or contents of existing AVPs, or checking the contents of the accounting record itself. One of these means could be to insert into the request sent to the accounting server an Auth-Application-Id AVP containing the identifier of the application for which the accounting request is sent. But in general, there is no clean and generic scheme for sorting these messages. Therefore, the use of this model is recommended only when all received accounting messages can be clearly identified and sorted. For most cases, the use of Coupled Accounting Model is recommended.

#### Coupled Accounting Model:

In this model, the accounting messages will use the Application Id of the application using the accounting service. The design implication for this is that the accounting messages are tightly coupled with the application itself; meaning that accounting messages will be routed like the other application messages. It would then be the responsibility of the application server (application entity receiving the ACR message) to send the accounting records carried by the accounting messages to the proper accounting server. The application server is also responsible for formulating a proper response (ACA). A coupled accounting model is a good design choice when:

- \* The system architecture or deployment does not provide an accounting server that supports Diameter. Consequently, the application server has to be provisioned to use a different protocol to access the accounting server, e.g., via LDAP, SOAP etc. This case includes the support of older accounting systems that are not Diameter aware.
- \* The system architecture or deployment requires that the accounting service for the specific application should be handled by the application itself.

In all cases above, there will generally be no direct Diameter access to the accounting server.

These models provide a basis for using accounting messages. Application designers may obviously deviate from these models provided that the factors being addressed here have also been taken into account. Although it is not recommended, an application may

define a new set of commands to carry application-specific accounting records.

#### 5.11. Diameter Security Mechanisms

As specified in [RFC6733], the Diameter message exchange should be secured between neighboring Diameter peers using TLS/TCP or DTLS/SCTP. However, IPsec can also be deployed to secure communication between Diameter peers. When IPsec is used instead of TLS or DTLS, the following recommendations apply.

IPsec ESP [RFC4301] in transport mode with non-null encryption and authentication algorithms is used to provide per-packet authentication, integrity protection and confidentiality, and support the replay protection mechanisms of IPsec. IKEv2 [RFC5996] is recommended for performing mutual authentication and for establishing and maintaining security associations (SAs).

IKEv1 [RFC2409] was used with RFC 3588 [RFC3588] and for easier migration from IKEv1 based implementations both RSA digital signatures and pre-shared keys should be supported in IKEv2. However, if IKEv1 is used, implementers should follow the guidelines given in Section 13.1 of RFC 3588 [RFC3588].

#### 6. Defining Generic Diameter Extensions

Generic Diameter extensions are AVPs, commands or applications that are designed to support other Diameter applications. They are auxiliary applications meant to improve or enhance the Diameter protocol itself or Diameter applications/functionality. Some examples include the extensions to support auditing and redundancy (see [I-D.calhoun-diameter-res-mgmt]), improvements in duplicate detection scheme (see [I-D.asveren-dime-dupcons]), and the support for QoS AVPs (see [RFC5777]).

Since generic extensions may cover many aspects of Diameter and Diameter applications, it is not possible to enumerate all scenarios. However, some of the most common considerations are as follows:

##### Backward Compatibility:

With the design of generic extensions an protocol designer has to consider with potential concerns about how existing applications deal with the new extension they do not understand. Designers also have to make sure that new extensions do not break expected message delivery layer behavior.

##### Forward Compatibility:

Protocol designers need to make sure that their design will not introduce undue restrictions for future applications.

#### Trade-off in Signaling:

Designers may have to choose between the use of optional AVPs piggybacked onto existing commands versus defining new commands and applications. Optional AVPs are simpler to implement and may not need changes to existing applications. However, this ties the sending of extension data to the application's transmission of a message. This has consequences if the application and the extensions have different timing requirements. The use of commands and applications solves this issue, but the trade-off is the additional complexity of defining and deploying a new application. It is left up to the designer to find a good balance among these trade-offs based on the requirements of the extension.

In practice, generic extensions often use optional AVPs because they are simple and non-intrusive to the application that would carry them. Peers that do not support the generic extensions need not understand nor recognize these optional AVPs. However, it is recommended that the authors of the extension specify the context or usage of the optional AVPs. As an example, in the case that the AVP can be used only by a specific set of applications then the specification must enumerate these applications and the scenarios when the optional AVPs will be used. In the case where the optional AVPs can be carried by any application, it should be sufficient to specify such a use case and perhaps provide specific examples of applications using them.

In most cases, these optional AVPs piggybacked by applications would be defined as a Grouped AVP and it would encapsulate all the functionality of the generic extension. In practice, it is not uncommon that the Grouped AVP will encapsulate an existing AVP that has previously been defined as mandatory ('M'-bit set) e.g., 3GPP IMS Cx/Dx interfaces ([TS29.228] and [TS29.229]).

## 7. Guidelines for Registrations of Diameter Values

As summarized in the Section 3 of this document and further described in the Section 1.3 of [RFC6733], there are four main ways to extend Diameter. The process for defining new functionality slightly varies based on the different extensions. This section provides protocol designers with some guidance regarding the definition of values for possible Diameter extensions and the necessary interaction with IANA to register the new functionality.

### a. Defining new AVP values

The specifications defining AVPs and AVP values provide guidance for defining new values and the corresponding policy for adding these values. For example, the RFC 5777 [RFC5777] defines the Treatment-Action AVP which contains a list of valid values corresponding to pre-defined actions (drop, shape, mark, permit). This set of values can be extended following the Specification Required policy defined in [RFC5226]. As a second example, the Diameter base specification [RFC6733] defines the Result-Code AVP that contains a 32-bit address space used to identify possible errors. According to the Section 11.3.2 of [RFC6733], new values can be assigned by IANA via an IETF Review process [RFC5226].

b. Creating new AVPs

Two different types of AVP Codes namespaces can be used to create a new AVPs:

- \* IETF AVP Codes namespace;
- \* Vendor-specific AVP Codes namespace.

In the latter case, a vendor needs to be first assigned by IANA with a private enterprise number, which can be used within the Vendor-Id field of the vendor-specific AVP. This enterprise number delimits a private namespace in which the vendor is responsible for vendor-specific AVP code value assignment. The absence of a Vendor-Id or a Vendor-Id value of zero (0) in the AVP header identifies standard AVPs from the IETF AVP Codes namespace managed by IANA. The allocation of code values from the IANA-managed namespace is conditioned by an Expert Review of the specification defining the AVPs or an IETF review if a block of AVPs needs to be assigned. Moreover, the remaining bits of the AVP Flags field of the AVP header can be also assigned via Standard Action if the creation of new AVP Flags is desired.

c. Creating new commands

Unlike the AVP Code namespace, the Command Code namespace is flat but the range of values is subdivided into three chunks with distinct IANA registration policies:

- \* A range of standard Command Code values that can be allocated via IETF review;
- \* A range of vendor-specific Command Code values that can be allocated on a First-Come/First-Served basis;

- \* A range of values reserved only for experimental and testing purposes.

As for AVP Flags, the remaining bits of the Command Flags field of the Diameter header can also be assigned via a Standards Action to create new Command Flags if required.

#### d. Creating new applications

Similarly to the Command Code namespace, the Application-Id namespace is flat but divided into two distinct ranges:

- \* A range of values reserved for standard Application-Ids allocated after Expert Review of the specification defining the standard application;
- \* A range for values for vendor specific applications, allocated by IANA on a First-Come/First-Serve basis.

The IANA AAA parameters page can be found at <http://www.iana.org/assignments/aaa-parameters/aaa-parameters.xml> and the enterprise number IANA page is available at <http://www.iana.org/assignments/enterprise-numbers>. More details on the policies followed by IANA for namespace management (e.g. First-Come/First-Served, Expert Review, IETF Review, etc.) can be found in [RFC5226].

#### NOTE:

When the same functionality/extension is used by more than one vendor, it is recommended to define a standard extension. Moreover, the registration of vendor-specific extension is encouraged to avoid interoperability issues in the same network. With this aim, the registration policy of vendor-specific extension has been simplified with the publication of [RFC6733] and the namespace reserved for vendor-specific extensions is large enough to avoid exhaustion.

#### 8. IANA Considerations

This document does not require actions by IANA.

#### 9. Security Considerations

This document provides guidelines and considerations for extending Diameter and Diameter applications. Although such an extension may related to a security functionality, the document does not explicitly give guidance on enhancing Diameter with respect to security.

## 10. Contributors

The content of this document was influenced by a design team created to revisit the Diameter extensibility rules. The team consisting of the members listed below was formed in February 2008 and finished its work in June 2008.

- o Avi Lior
- o Glen Zorn
- o Jari Arkko
- o Lionel Morand
- o Mark Jones
- o Victor Fajardo
- o Tolga Asveren
- o Jouni Korhonen
- o Glenn McGregor
- o Hannes Tschofenig
- o Dave Frascone

We would like to thank Tolga Asveren, Glenn McGregor, and John Loughney for their contributions as co-authors to earlier versions of this document.

## 11. Acknowledgments

We greatly appreciate the insight provided by Diameter implementers who have highlighted the issues and concerns being addressed by this document. The authors would also like to thank Jean Mahoney, Ben Campbell and Sebastien Decugis for their invaluable detailed reviews and comments on this document.

## 12. Informative References

- [I-D.asveren-dime-dupcons]  
Asveren, T., "Diameter Duplicate Detection Cons.", draft-asveren-dime-dupcons-00 (work in progress), August 2006.
- [I-D.calhoun-diameter-res-mgmt]



Calhoun, P., "Diameter Resource Management Extensions", draft-calhoun-diameter-res-mgmt-08.txt (work in progress), March 2001.

[Q.3303.3]

3rd Generation Partnership Project, "ITU-T Recommendation Q.3303.3, "Resource control protocol no. 3 (rcp3): Protocol at the Rw interface between the Policy Decision Physical Entity (PD-PE) and the Policy Enforcement Physical Entity (PE-PE): Diameter"", 2008.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[RFC3588] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[RFC4005] Calhoun, P., Zorn, G., Spence, D., and D. Mitton, "Diameter Network Access Server Application", RFC 4005, August 2005.

[RFC4072] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4740] Garcia-Martin, M., Belinchon, M., Pallares-Lopez, M., Canales-Valenzuela, C., and K. Tammi, "Diameter Session Initiation Protocol (SIP) Application", RFC 4740, November 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5447] Korhonen, J., Bournelle, J., Tschofenig, H., Perkins, C., and K. Chowdhury, "Diameter Mobile IPv6: Support for Network Access Server to Diameter Server Interaction", RFC 5447, February 2009.

[RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, February 2010.

- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen,  
"Internet Key Exchange Protocol Version 2 (IKEv2)", RFC  
5996, September 2010.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn,  
"Diameter Base Protocol", RFC 6733, October 2012.
- [TS29.228]  
3rd Generation Partnership Project, "3GPP TS 29.228;  
Technical Specification Group Core Network and Terminals;  
IP Multimedia (IM) Subsystem Cx and Dx Interfaces;  
Signalling flows and message contents",  
<<http://www.3gpp.org/ftp/Specs/html-info/29272.htm>>.
- [TS29.229]  
3rd Generation Partnership Project, "3GPP TS 29.229;  
Technical Specification Group Core Network and Terminals;  
Cx and Dx interfaces based on the Diameter protocol;  
Protocol details",  
<<http://www.3gpp.org/ftp/Specs/html-info/29229.htm>>.
- [TS29.328]  
3rd Generation Partnership Project, "3GPP TS 29.328;  
Technical Specification Group Core Network and Terminals;  
IP Multimedia (IM) Subsystem Sh interface; signalling  
flows and message content",  
<<http://www.3gpp.org/ftp/Specs/html-info/29328.htm>>.
- [TS29.329]  
3rd Generation Partnership Project, "3GPP TS 29.329;  
Technical Specification Group Core Network and Terminals;  
Sh Interface based on the Diameter protocol; Protocol  
details",  
<<http://www.3gpp.org/ftp/Specs/html-info/29329.htm>>.

#### Authors' Addresses

Lionel Morand (editor)  
Orange Labs  
38/40 rue du General Leclerc  
Issy-Les-Moulineaux Cedex 9 92794  
France

Phone: +33145296257  
Email: [lionel.morand@orange.com](mailto:lionel.morand@orange.com)

Victor Fajardo

Email: [vf0213@gmail.com](mailto:vf0213@gmail.com)

Hannes Tschofenig  
Nokia Siemens Networks  
Linnoitustie 6  
Espoo 02600  
Finland

Phone: +358 (50) 4871445  
Email: [Hannes.Tschofenig@gmx.net](mailto:Hannes.Tschofenig@gmx.net)  
URI: <http://www.tschofenig.priv.at>

Diameter Maintenance and Extensions (DIME)  
Internet-Draft  
Intended Status: Proposed Standard  
Expires: October 15, 2014

B. Hirschman  
L. Bertz  
Sprint  
April 2014

Diameter Congestion and Filter Attributes  
draft-ietf-dime-congestion-flow-attributes-00.txt

Abstract

This document defines optional ECN and filter related attributes that can be used for improved traffic identification, support of ECN and minimized filter administration within Diameter.

RFC 5777 defines a Filter-Rule AVP that accommodates extensions for classification, conditions and actions. It does not support traffic identification for packets using Explicit Congestion Notification as defined in RFC 3168 and does not provide specific actions when the flow(s) described by the Filter-Rule are congested.

A Filter-Rule can describe multiple flows but not the exact number of flows. Flow count and other associated data (e.g. packets) is not captured in Accounting applications, leaving administrators without useful information regarding the effectiveness or understanding of the filter definition.

These optional attributes are forward and backwards compatible with RFC 5777.

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 February 14, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	4
2. Terminology and Abbreviations . . . . .	4
3. ECN-IP-Codepoint, Congestion-Treatment and Filter Attributes .	4
3.1. ECN-IP-Codepoint AVP . . . . .	4
3.2. Congestion-Treatment AVP . . . . .	5
3.3. Flow-Count AVP . . . . .	5
3.4. Packet-Count AVP . . . . .	5
4. IANA Considerations . . . . .	5
4.1. AVP Codes . . . . .	5
5. Security Considerations . . . . .	6
6. Acknowledgements . . . . .	6
7. References . . . . .	6
7.1. Normative References . . . . .	6
Authors' Addresses . . . . .	6

## 1. Introduction

Two optional Explicit Congestion Notification (ECN) [RFC3168] related AVPs are specified in the document. The first AVP provides direct support for ECN [RFC3168] in the IP header and the second AVP provides the ability to define alternate traffic treatment when congestion is experienced.

This document also defines two optional AVPs, Flow-Count and Packet-Count, used for conveying flow information within the Diameter protocol [RFC6733]. These AVPs were found to be useful for a wide range of applications. The AVPs provide a way to convey information of the group of flows described by the Filter-Rule, IPFilterRule or other Diameter traffic filters.

The semantics and encoding of all AVPs can be found in Section 3.

Such AVPs are, for example, needed by some ECN applications to determine the number of flows congested or used by administrators to determine the impact of filter definitions.

Additional parameters may be defined in future documents as the need arises. All parameters are defined as Diameter-encoded Attribute Value Pairs (AVPs), which are described using a modified version of the Augmented Backus-Naur Form (ABNF), see [RFC6733]. The data types are also taken from [RFC6733].

## 2. Terminology and Abbreviations

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 [RFC2119].

## 3. ECN-IP-Codepoint, Congestion-Treatment and Filter Attributes

### 3.1. ECN-IP-Codepoint AVP

The ECN-IP-Codepoint AVP (AVP Code TBD) is of type Enumerated and specifies the Explicit Congestion Notification codepoint values to match in the IP header.

Value	Binary	Keyword	References
0	00	Non-ECT (Not ECN-Capable Transport)	[RFC3168]
1	01	ECT(1) (ECN-Capable Transport)	[RFC3168]
2	10	ECT(0) (ECN-Capable Transport)	[RFC3168]
3	11	CE (Congestion Experienced)	[RFC3168]

When this AVP is used for classification in the Filter-Rule it MUST be part of Classifier Grouped AVP as defined in RFC5777.

### 3.2. Congestion-Treatment AVP

The Congestion-Treatment AVP (AVP Code TBD) is of type Grouped and indicates how congested traffic, i.e., traffic that has Explicit Congestion Notification Congestion Experienced marking set or some other administratively defined criteria, is treated. In case the Congestion-Treatment AVP is absent the treatment of the congested traffic is left to the discretion of the node performing QoS treatment.

```

Congestion-Treatment ::= < AVP Header: TBD >
                        { Treatment-Action }
                        [ QoS-Profile-Template ]
                        [ QoS-Parameters ]
                        * [ AVP ]

```

Treatment-Action, QoS-Profile-Template and QoS-Parameters are defined in [RFC5777]. The Congestion-Treatment AVP is an action and MUST be an attribute of the Filter-Rule Grouped AVP as defined in RFC5777.

### 3.3. Flow-Count AVP

The Flow-Count AVP (AVP Code TBD) is of type Unsigned64.

It indicates the number of protocol specific flows. The protocol is determined by the filter (e.g. IPFilterRule, Filter-Id, etc.).

### 3.4. Packet-Count AVP

The Packet-Count AVP (AVP Code TBD) is of type Unsigned64.

It indicates the number of protocol specific packets. The protocol is determined by the filter (e.g. IPFilterRule, Filter-Id, etc.).

## 4. IANA Considerations

### 4.1. AVP Codes

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		Code	Defined	Section	
					Type



ECN-IP-Codepoint	TBD 3.1	Enumerated
Congestion-Treatment	TBD 3.2	Grouped
Flow-Count	TBD 3.3	Unsigned64
Packet-Count	TBD 3.4	Unsigned64

## 5. Security Considerations

The document does not raise any new security concerns. This document describes an extension of RFC5777 that introduces a new filter parameter applied to ECN as defined by [RFC3168]. It also defines a new Grouped AVP that expresses what action to take should congestion be detected. The Grouped AVP reuses attributes defined in RFC5777.

The security considerations of the Diameter protocol itself have been discussed in RFC 6733 [RFC6733]. Use of the AVPs defined in this document MUST take into consideration the security issues and requirements of the Diameter base protocol.

## 6. Acknowledgements

We would like to thank Avi Lior for his guidance and feedback during the development of this specification.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3168] Black, D., Floyd, S., and K. Ramakrishnan, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Lior, A. and Jones, M. Ed., "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, February 2010.

### Authors' Addresses

Lyle Bertz  
Sprint

6220 Sprint Parkway  
Overland Park, KS 66251  
United States

EMail: Lyle.T.Bertz@sprint.com

Brent Hirschman  
Sprint  
6220 Sprint Parkway  
Overland Park, KS 66251  
United States

EMail: Brent.Hirschman@sprint.com

Diameter Maintenance and Extensions (DIME)  
Internet-Draft  
Intended Status: Proposed Standard  
Expires: December 1, 2015

L. Bertz  
S. Manning  
Sprint  
B. Hirschman  
July 2015

Diameter Congestion and Filter Attributes  
draft-ietf-dime-congestion-flow-attributes-02.txt

Abstract

This document defines optional Diameter attributes that can be used to help manage networks that use Explicit Congestion Notification (ECN) or Diameter traffic filters. These new attributes allow for improved data traffic identification, support of ECN and minimize Diameter filter administration.

RFC 5777 defines a Filter-Rule Attribute Value Pair (AVP) that accommodates extensions for classification, conditions and actions. It however, does not support traffic identification for packets using Explicit Congestion Notification as defined in RFC 3168 and does not provide specific actions when the flow(s) described by the Filter-Rule are congested.

Further, a Filter-Rule can describe multiple flows but not the exact number of flows. Flow count and other associated data (e.g. packets) are not captured by Accounting applications, leaving administrators without useful information regarding the effectiveness or appropriateness of the filter definition.

The optional attributes defined in this document are forward and backwards compatible with RFC 5777.

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 1, 2015.

#### Copyright Notice

Copyright (c) 2015 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 . . . . .	4
2. Terminology and Abbreviations . . . . .	4
3. ECN-IP-Codepoint, Congestion-Treatment and Filter Attributes .	4
3.1. ECN-IP-Codepoint AVP . . . . .	4
3.2. Congestion-Treatment AVP . . . . .	5
3.3. Flow-Count AVP . . . . .	5
3.4. Packet-Count AVP . . . . .	5
4. IANA Considerations . . . . .	6
4.1. AVP Codes . . . . .	6
5. Examples . . . . .	6
5.1. Classifier Example . . . . .	6
5.2. Diameter Credit Control (CC) with Congestion Information .	7
6. Security Considerations . . . . .	8
7. Acknowledgements . . . . .	9
8. References . . . . .	9
8.1. Normative References . . . . .	9
Authors' Addresses . . . . .	9

## 1. Introduction

Two optional Explicit Congestion Notification (ECN) [RFC3168] related AVPs are specified in this document. The first AVP provides direct support for filtering ECN marked traffic [RFC3168] and the second AVP provides the ability to define alternate traffic treatment when congestion is experienced.

This document also defines two optional AVPs, Flow-Count and Packet-Count, used for conveying flow information within the Diameter protocol [RFC6733]. These AVPs were found to be useful for a wide range of applications. The AVPs provide a way to convey information of the group of flows described by the Filter-Rule, IPFilterRule or other Diameter traffic filters.

The semantics and encoding of all AVPs can be found in Section 3.

Such AVPs are, for example, needed by some congestion management functions to determine the number of flows congested or used by administrators to determine the impact of filter definitions.

Additional parameters may be defined in future documents as the need arises. All parameters are defined as Diameter-encoded Attribute Value Pairs (AVPs), which are described using a modified version of the Augmented Backus-Naur Form (ABNF), see [RFC6733]. The data types are also taken from [RFC6733].

## 2. Terminology and Abbreviations

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 [RFC2119].

## 3. ECN-IP-Codepoint, Congestion-Treatment and Filter Attributes

### 3.1. ECN-IP-Codepoint AVP

The ECN-IP-Codepoint AVP (AVP Code TBD1) is of type Enumerated and specifies the Explicit Congestion Notification codepoint values to match in the IP header.

Value	Binary	Keyword	References
0	00	Non-ECT (Not ECN-Capable Transport)	[RFC3168]
1	01	ECT(1) (ECN-Capable Transport)	[RFC3168]
2	10	ECT(0) (ECN-Capable Transport)	[RFC3168]
3	11	CE (Congestion Experienced)	[RFC3168]

When this AVP is used for classification in the Filter-Rule it MUST be part of Classifier Grouped AVP as defined in RFC5777.

### 3.2. Congestion-Treatment AVP

The Congestion-Treatment AVP (AVP Code TBD2) is of type Grouped. It indicates how to treat traffic IP (5-tuple) flow(s) when congestion is detected. The detection of congestion can be based on the reception of IP packets with the Congestion Experience (CE) codepoint set (see [RFC3168]) or by any other administratively defined criteria.

A Filter-Rule may contain a Classifier that describes one or many 5-tuples per RFC5777. This treatment applies to all packets associated to all 5-tuples (flows) captured by the Filter-Rule.

If the Congestion-Treatment AVP is absent the treatment of the congested traffic is left to the discretion of the node performing QoS treatment.

```
Congestion-Treatment ::= < AVP Header: TBD2 >
                        { Treatment-Action }
                        [ QoS-Profile-Template ]
                        [ QoS-Parameters ]
                        * [ AVP ]
```

Treatment-Action, QoS-Profile-Template and QoS-Parameters are defined in RFC5777. The Congestion-Treatment AVP is an action and MUST be an attribute of the Filter-Rule Grouped AVP as defined in RFC5777.

### 3.3. Flow-Count AVP

The Flow-Count AVP (AVP Code TBD3) is of type Unsigned64.

It indicates the number of protocol specific flows. The protocol is determined by the filter (e.g. IPFilterRule, Filter-Id, etc.).

### 3.4. Packet-Count AVP

The Packet-Count AVP (AVP Code TBD4) is of type Unsigned64.

It indicates the number of protocol specific packets. The protocol is determined by the filter (e.g. IPFilterRule, Filter-Id, etc.).

## 4. IANA Considerations

### 4.1. AVP Codes

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
ECN-IP-Codepoint	TBD1	3.1	Enumerated
Congestion-Treatment	TBD2	3.2	Grouped
Flow-Count	TBD3	3.3	Unsigned64
Packet-Count	TBD4	3.4	Unsigned64

## 5. Examples

The following examples illustrate the use of the AVPs defined in this draft.

### 5.1. Classifier Example

The Classifier AVP (AVP Code 511) specified in RFC5777 is a grouped AVP that consists of a set of attributes that specify how to match a packet. The addition of the ECN-IP-Codepoint is shown here.

```
Classifier ::= < AVP Header: 511 >
  { Classifier-ID }
  [ Protocol ]
  [ Direction ]
  [ ECN-IP-Codepoint ]
  * [ From-Spec ]
  * [ To-Spec ]
  * [ Diffserv-Code-Point ]
  [ Fragmentation-Flag ]
  * [ IP-Option ]
  * [ TCP-Option ]
  [ TCP-Flags ]
  * [ ICMP-Type ]
  * [ ETH-Option ]
  * [ AVP ]
```

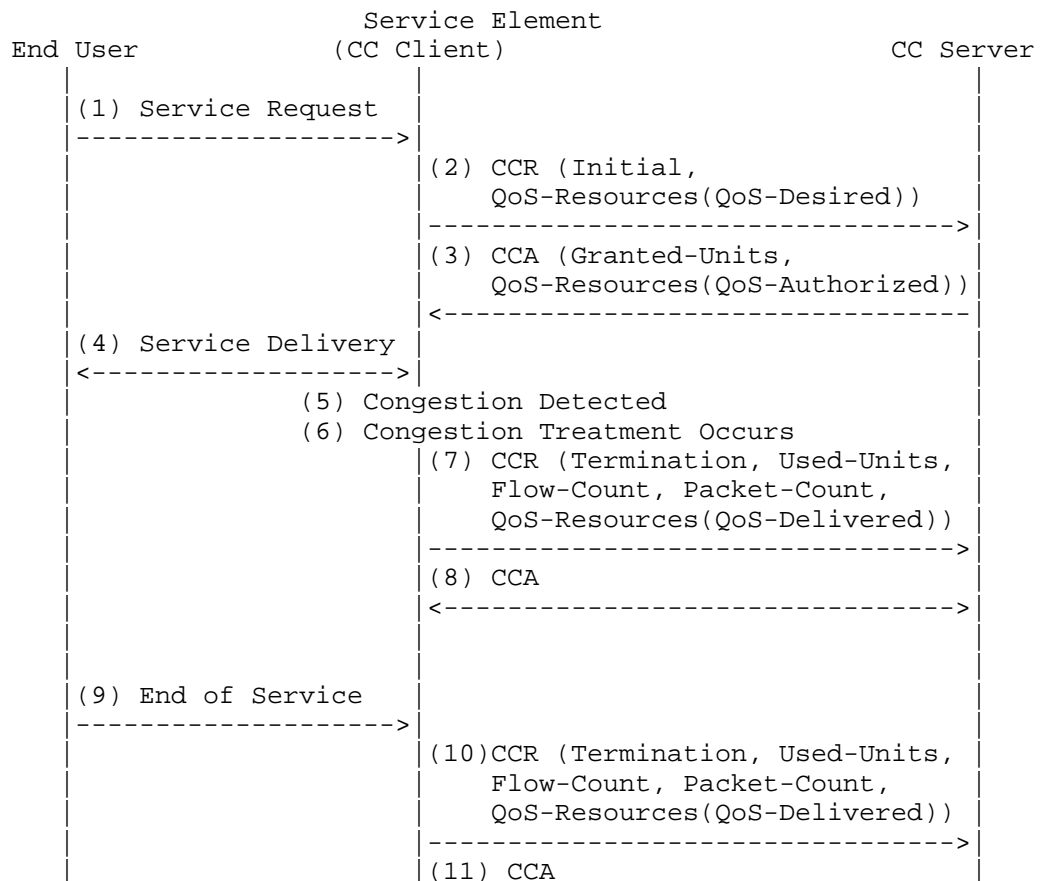
Setting the ECN-IP-Codepoint value to 'CE' would permit the capture of CE flags in the Flow.



Another Classifier with the ECN-IP-Codepoint value of 'ECT' could be specified and, when coupled with the Flow-Count AVP, reports the number of ECT capable flows.

## 5.2. Diameter Credit Control (CC) with Congestion Information

Diameter nodes using Credit Control can use the Congestion-Treatment AVP to trigger specific actions when congestion occurs. This is similar to the Excess-Treatment Action. The ability to detect when congestion occurs is specific to the AVPs in the Filter-Rule and Diameter Client and is no different than how 'Excess' can be determined for Excess-Treatment. If conditions associated with Excess-Treatment [RFC5777] or Congestion-Treatment has occurred Diameter Clients may autonomously send Credit Control Requests (CCRs) during the Service Delivery session as interim events. This is shown in Figure 1.



| |<-----|

Figure 1: Example of a Diameter Credit Control with Congestion Information

The 'Used-Service-Units' described in RFC5777 examples is customarily a Service-Units, Time-Units or Byte-Count AVP. This is insufficient to represent network state and does not differentiate between throughput and good-put (good or quality throughput) even though the filters may imply good or poor throughput.

Flow-Count and Packet-Count AVPs defined in this document could be sent with a CCR when the triggering event is related to Congestion-Treatment. This provides the CC Server with a better view of the type of congested traffic for improved decision making and charging. Sending such AVPs under any condition permits rudimentary traffic profiling regardless of network conditions. For instance, low byte per packet counts is indicative of web traffic and high byte counts per packet with a small number of flows may be indicative of video traffic. Enriched reporting described here provides relief from Deep Packet Inspection load and loss of information as traffic becomes increasingly encrypted.

Some services, e.g. Streaming Services, limit the number of flows, Flow-Count, as opposed to other Units, i.e. Byte-Count. In such a case the Flow-Count AVP may be used in place of Service-Units.

## 6. Security Considerations

This document describes an extension of RFC5777 that introduces a new filter parameter applied to ECN as defined by [RFC3168]. It also defines a new Grouped AVP that expresses what action to take should congestion be detected. The Grouped AVP reuses attributes defined in RFC5777. As these are extensions to RFC 5777, they do not raise new security concerns.

The Flow-Count and Packet-Count AVPs can be provided in conjunction with customary AVPs, e.g. Bytes, Time, Service Units, during Accounting activities as described in the base protocol [RFC6733] or other Diameter applications. These new AVPs provide more information that can be privacy sensitive. The privacy sensitivity is directly related to traffic captured by Filters and associated reports. Narrow filtering, which creates the highest level of privacy sensitivity, is too resource intensive to be widely applied on large networks. Paradoxically, improving reporting information lessens the depth of inspection required to characterize traffic for many congestion management activities as noted in Section 5.2.

If an administrator can provide congestion actions without the need to report them to a Diameter application they should use the Congestion-Treatment AVP which also reduces Diameter traffic during congestion events.

The security considerations of the Diameter protocol itself have been discussed in RFC 6733 [RFC6733]. Use of the AVPs defined in this document MUST take into consideration the security issues and requirements of the Diameter base protocol.

## 7. Acknowledgements

We would like to thank Avi Lior for his guidance and feedback during the development of this specification.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3168] Black, D., Floyd, S., and K. Ramakrishnan, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Lior, A. and Jones, M. Ed., "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, February 2010.

### Authors' Addresses

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

EMail: [lyleb551144@gmail.com](mailto:lyleb551144@gmail.com)

Serge Manning  
Sprint  
6220 Sprint Parkway  
Overland Park, KS 66251

United States

EMail: sergem913@gmail.com

Brent Hirschman

EMail: Brent.Hirschman@gmail.com

DIME  
Internet-Draft  
Intended status: Informational  
Expires: April 24, 2014

H. Tschofenig, Ed.  
Nokia Solutions and Networks  
J. Korhonen  
Broadcom  
G. Zorn  
Network Zen  
K. Pillay  
Oracle Communications  
October 21, 2013

Diameter AVP Level Security: Scenarios and Requirements  
draft-ietf-dime-e2e-sec-req-01.txt

## Abstract

This specification discusses requirements for providing Diameter security at the level of individual Attribute Value Pairs.

## 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 April 24, 2014.

## Copyright Notice

Copyright (c) 2013 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 . . . . .	2
2. Terminology . . . . .	3
3. Security Threats . . . . .	3
4. Scenarios for Diameter AVP-Level Protection . . . . .	4
5. Requirements . . . . .	6
6. Open Issues . . . . .	7
7. Security Considerations . . . . .	8
8. IANA Considerations . . . . .	8
9. Acknowledgments . . . . .	8
10. References . . . . .	8
10.1. Normative References . . . . .	8
10.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

The Diameter Base specification [2] offers security protection between neighboring Diameter peers and mandates that either TLS (for TCP), DTLS (for SCTP), or IPsec is used. These security protocols offer a wide range of security properties, including entity authentication, data-origin authentication, integrity, confidentiality protection and replay protection. They also support a large number of cryptographic algorithms, algorithm negotiation, and different types of credentials.

The need to also offer additional security protection of AVPs between non-neighboring Diameter nodes was recognized very early in the work on Diameter. This lead to work on Diameter security using the Cryptographic Message Syntax (CMS) [3]. Due to lack of deployment interest at that time (and the complexity of the developed solution) the specification was, however, never completed.

In the meanwhile Diameter had received a lot of deployment interest from the cellular operator community and because of the sophistication of those deployments the need for protecting Diameter AVPs between non-neighboring nodes re-surfaced. Since early 2000 (when the work on [3] was discontinued) the Internet community had seen advances in cryptographic algorithms (for example, authenticated encryption algorithms) and new security building blocks were developed.

This document collects requirements for developing a solution to protect Diameter AVPs.

## 2. Terminology

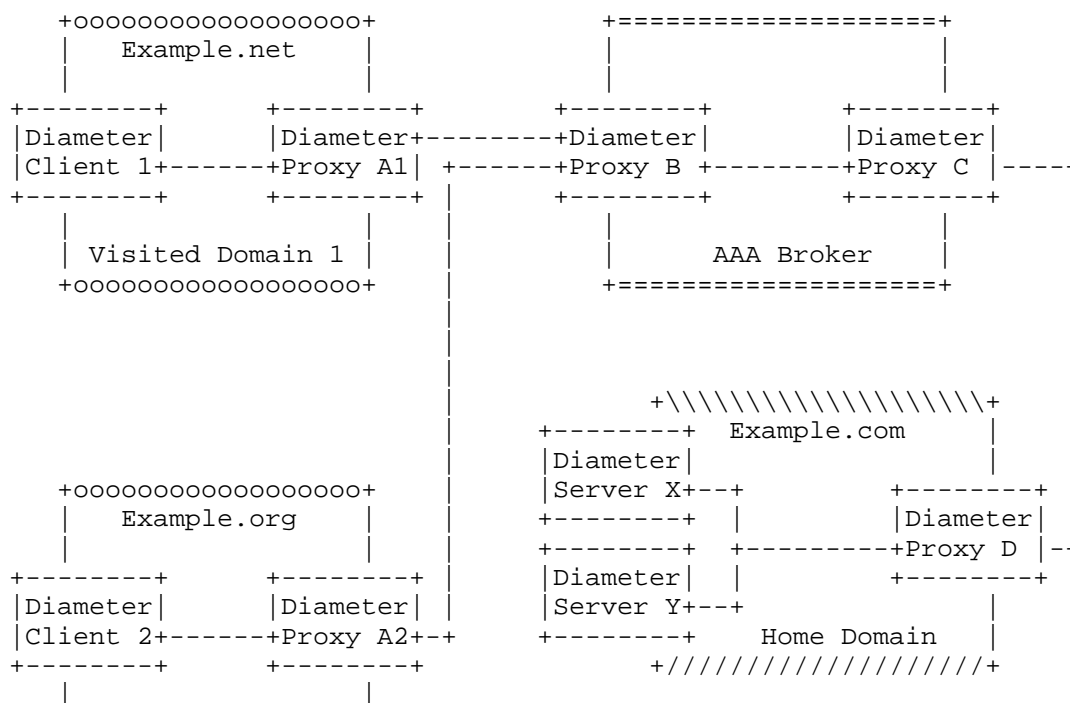
The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL' in this specification are to be interpreted as described in [1].

This document re-uses terminology from the Diameter base specification [2].

In the figures below we use the symbols 'AVP' and '{AVP}k'. AVP refers to an unprotected AVP and {AVP}k refers to an AVP that experiences security protection (using key "k") without further distinguishing between integrity and confidentiality protection.

## 3. Security Threats

The follow description aims to illustrate various security threats that raise the need for protecting Diameter Attribute Value Pairs (AVPs). Figure 1 illustrates an example Diameter topology where a Diameter clients want to interact with the example.com home domain. To interconnect the two visited networks a AAA interconnection provider, labeled as AAA Broker, is used.



```
| Visited Domain 2 |
+oooooooooooooooooooo+
```

Figure 1: Example Diameter Deployment.

**Eavesdropping:** Some Diameter applications carry information that is only intended for consumption by end points, either by the Diameter client or by the Diameter server but not by intermediaries. As an example, consider the Diameter EAP application [4] that allows keying material for the protection of air interface between the end device and the network access server to be carried from the Diameter server to the Diameter client (using the EAP-Master-Session-Key AVP). The content of the EAP-Master-Session-Key AVP would benefit from protection against eavesdropping by intermediaries. Other AVPs might also carry sensitive personal data that, when collected by intermediaries, allow for traffic analysis.

In context of the deployment shown in Figure 1 the adversary could, for example, be in the AAA broker network.

**Injection and Manipulation:** The Diameter base specification mandates security protection between neighboring nodes but Diameter agents may be compromised or misconfigured and inject/manipulate AVPs. To detect such actions additional security protection needs to be applied at the Diameter layer.

Nodes that could launch such an attack are any Diameter agents along the end-to-end communication path.

**Impersonation:** Imagine a case where a Diameter message from Example.net contains information claiming to be from Example.org. This would either require strict verification at the edge of the AAA broker network or cryptographic assurance at the Diameter layer to prevent a successful impersonation attack.

Any Diameter realm could launch such an attack aiming for financial benefits or to disrupt service availability.

#### 4. Scenarios for Diameter AVP-Level Protection

This scenario outlines a number of cases for deploying security protection of individual Diameter AVPs.



In the first scenario, shown in Figure 2, end-to-end security protection is provided between the Diameter client and the Diameter server. Diameter AVPs exchanged between these two Diameter nodes are protected.



Figure 2: End-to-End Diameter AVP Security Protection.

In the second scenario, shown in Figure 3, a Diameter proxy acts on behalf of the Diameter client with regard to security protection. It applies security protection to outgoing Diameter AVPs and verifies incoming AVPs.

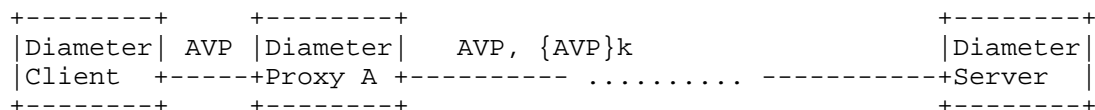


Figure 3: Middle-to-End Diameter AVP Security Protection.

In the third scenario shown in Figure 4 a Diameter proxy acts on behalf of the Diameter server.

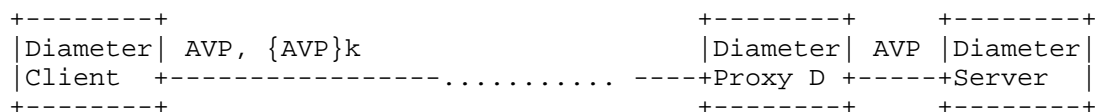


Figure 4: End-to-Middle Diameter AVP Security Protection.

The fourth and the final scenario (see Figure 5) is a combination of the end-to-middle and the middle-to-end scenario shown in Figure 4 and in Figure 3. From a deployment point of view this scenario is easier to accomplish for two reasons: First, Diameter clients and Diameter servers remain unmodified. This ensures that no modifications are needed to the installed Diameter infrastructure. Second, key management is also simplified since fewer number of key pairs need to be negotiated and provisioned.

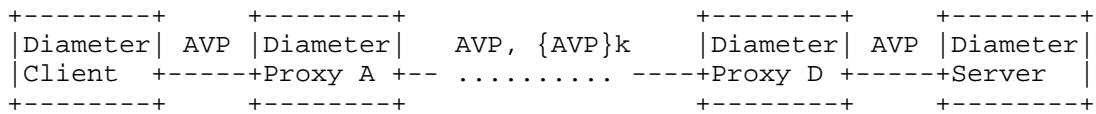


Figure 5: Middle-to-Middle Diameter AVP Security Protection.

Various security threats are mitigated by selectively applying security protection for individual Diameter AVPs. Without protection there is the possibility for password sniffing, confidentiality violation, AVP insertion, deletion or modification. Additionally, applying digital signature offers non-repudiation capabilities; a feature not yet available in today's Diameter deployment. Modification of certain Diameter AVPs may not necessarily be the act of malicious behavior but could also be the result of misconfiguration. An over-aggressively configured firewalling Diameter proxy may also remove certain AVPs. In most cases data origin authentication and integrity protection of AVPs will provide most benefits for existing deployments with minimal overhead and (potentially) operating in a full-backwards compatible manner.

## 5. Requirements

Requirement #1: Solutions MUST support an extensible set of cryptographic algorithms.

Motivation: Crypto-agility is the ability of a protocol to adapt to evolving cryptographic algorithms and security requirements. This may include the provision of a modular mechanism to allow cryptographic algorithms to be updated without substantial disruption to deployed implementations.

Requirement #2: Solutions MUST support confidentiality, integrity, and data-origin authentication. Solutions for integrity protection MUST work in a backwards-compatible way with existing Diameter applications.

Requirement #3: Solutions MUST support replay protection. Any Diameter node has an access to network time and thus can synchronise their clocks.

Requirement #4: Solutions MUST support the ability to delegate security functionality to another entity

Motivation: As described in Section 4 the ability to let a Diameter proxy to perform security services on behalf of all clients within the same administrative domain is important for

incremental deployability. The same applies to the other communication side where a load balancer terminates security services for the servers it interfaces.

Requirement #5: Solutions MUST be able to selectively apply their cryptographic protection to certain Diameter AVPs.

Motivation: Some Diameter applications assume that certain AVPs are added, removed, or modified by intermediaries. As such, it MUST be possible to apply security protection selectively.

Requirement #6: Solutions MUST recommend a mandatory-to-implement cryptographic algorithm.

Motivation: For interoperability purposes it is beneficial to have a mandatory-to-implement cryptographic algorithm specified (unless profiles for specific usage environments specify otherwise).

Requirement #7: Solutions MUST support symmetric keys and asymmetric keys.

Motivation: Symmetric and asymmetric cryptographic algorithms provide different security services. Asymmetric algorithms, for example, allow non-repudiation services to be offered.

Requirement #8: A solution for dynamic key management MUST be included in the overall solution framework. However, it is assumed that no "new" key management protocol needs to be developed; instead existing ones are re-used, if at all possible. Rekeying could be triggered by (a) management actions and (b) expiring keying material.

Requirement #9: The ability to statically provisioned keys (symmetric as well as asymmetric keys) has to be supported to simplify management for small-scale deployments that typically do not have a back-end network management infrastructure.

## 6. Open Issues

Open Issue #1: Capability/Policy Discovery: This document talks about selectively protecting Diameter AVPs between different Diameter nodes. A Diameter node has to be configured such that it applies security protection to a certain number of AVPs. A number of policy related questions arise: What keying material should be used so that the intended recipient is also able to verify it? What AVPs shall be protected so that the result is not rejected by the recipient? In case of confidentiality protection the Diameter

node encrypting AVPs needs to know ahead of time what other node is intended to decrypt them. Should the list of integrity protected AVP be indicated in the protected payload itself (or is it known based on out-of-band information)? Is this policy / capability information assumed to be established out-of-band (manually) or is there a protocol mechanism to distribute this information?

Open Issue #2: Command-Line Support: Should solutions allow the provisioning of long-term shared symmetric credentials via a command-line interface / text file? This allows easier management for small-scale deployments.

## 7. Security Considerations

This entire document focused on the discussion of new functionality for securing Diameter AVPs selectively between non-neighboring nodes.

## 8. IANA Considerations

This document does not require actions by IANA.

## 9. Acknowledgments

We would like to thank Guenther Horn, Martin Dolly, for his review comments.

## 10. References

### 10.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

### 10.2. Informative References

- [3] Calhoun, P., Farrell, S., and W. Bulley, "Diameter CMS Security Application", draft-ietf-aaa-diameter-cms-sec-04 (work in progress), March 2002.
- [4] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.

## Authors' Addresses

Hannes Tschofenig (editor)  
Nokia Solutions and Networks  
Linnoitustie 6  
Espoo 02600  
Finland

Phone: +358 (50) 4871445  
Email: Hannes.Tschofenig@gmx.net  
URI: <http://www.tschofenig.priv.at>

Jouni Korhonen  
Broadcom  
Porkkalankatu 24  
Helsinki 00180  
Finland

Email: [jouni.nospam@gmail.com](mailto:jouni.nospam@gmail.com)

Glen Zorn  
Network Zen  
227/358 Thanon Sanphawut  
Bang Na Bangkok 10260  
Thailand

Email: [glenzorn@gmail.com](mailto:glenzorn@gmail.com)

Kervin Pillay  
Oracle Communications  
100 Crosby Drive  
Bedford, Massachusetts 01730  
USA

Email: [kervin.pillay@oracle.com](mailto:kervin.pillay@oracle.com)

Diameter Maintenance and Extensions (DIME)  
Internet-Draft  
Intended status: Standards Track  
Expires: January 5, 2015

M. Jones  
M. Liebsch  
L. Morand

July 4, 2014

Diameter Group Signaling  
draft-ietf-dime-group-signaling-04.txt

Abstract

In large network deployments, a single Diameter peer can support over a million concurrent Diameter sessions. Recent use cases have revealed the need for Diameter peers 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 peer 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 January 5, 2015.

Copyright Notice

Copyright (c) 2014 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 . . . . .	6
4.1.1. Group assignment at session initiation . . . . .	7
4.1.2. Removing a session from a session group . . . . .	8
4.1.3. Mid-session group assignment modifications . . . . .	9
4.2. Session Grouping Capability Discovery . . . . .	10
4.2.1. Explicit Capability Discovery . . . . .	10
4.2.2. Implicit Capability Discovery . . . . .	11
4.3. Deleting a Session Group . . . . .	11
4.4. Performing Group Operations . . . . .	11
4.4.1. Sending Group Commands . . . . .	11
4.4.2. Receiving Group Commands . . . . .	12
4.4.3. Error Handling for Group Commands . . . . .	12
4.4.4. Single-Session Fallback . . . . .	13
5. Operation with Proxies Agents . . . . .	13
6. Commands Formatting . . . . .	13
6.1. Formatting Example: Group Re-Auth-Request . . . . .	14
7. Attribute-Value-Pairs (AVP) . . . . .	14
7.1. Session-Group-Info AVP . . . . .	15
7.2. Session-Group-Control-Vector AVP . . . . .	15
7.3. Session-Group-Id AVP . . . . .	16
7.4. Group-Response-Action AVP . . . . .	16
7.5. Session-Group-Capability-Vector AVP . . . . .	17
8. Result-Code AVP Values . . . . .	17
9. IANA Considerations . . . . .	17
9.1. AVP Codes . . . . .	17
10. Security Considerations . . . . .	17
11. Acknowledgments . . . . .	18
12. Normative References . . . . .	18
Appendix A. Session Management -- Exemplary Session State	

Machines . . . . .	18
A.1. Authorization Session State Machine . . . . .	18
Authors' Addresses . . . . .	22

## 1. Introduction

In large network deployments, a single Diameter peer can support over a million concurrent Diameter sessions. Recent use cases have revealed the need for Diameter peers 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 peer 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 [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 peer 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 peer 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 peer 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 peer 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.

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 (peer 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 peer created the assignment	X	X
Remove a Session from a Session Group where the peer did not create the assignment		
Overrule a peer's group assignment *)		
Delete a Session Group owned by the peer	X	X
Delete a Session Group not owned by the peer		

#### Default Permission as per this Specification

\*) Editors' note: The protocol specification in this document does not consider overruling a peer's assignment of a session to a session group. Group signaling enabled applications may take such protocol support and associated protocol semantics into account in their specification.

## 4. Protocol Description

### 4.1. Session Grouping

Either Diameter peer 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 peer. Diameter AAA applications typically assign client and server roles to the Diameter peers, which are referred to as relevant Diameter peers 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 peers, 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 peer, each group pointing to individual sessions being assigned to the group. A peer must also keep a record about sessions, which have been assigned to a session group by that peer.

#### 4.1.1.1. 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 AAR [RFC4005], containing one or more group identifiers. Each of these groups need to 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 sessions from a list of existing session groups. Alternatively, the client may decide to create a new group and identify itself in the DiameterIdentity element of the Group-Session-Id AVP as per Section 7.3

The client MUST set the SESSION\_GROUP\_ALLOCATION\_ACTION of the Session-Group-Control-Vector AVP in each appended Session-Group-Info AVP to indicate that the identified session should be assigned to the identified session group.

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 of the Session-Group-Control-Vector AVP set, the server must assign the new session to each of the one or multiple identified session groups. In case one or multiple identified session groups are not known to the server, the server must add the one or multiple new groups 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 AAA [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 case, the server adds to the response additional Session-Group-Info AVPs, each identifying a session group, to which the server has assigned the new session. Each of the Session-Group-Info AVP added by the server must have the SESSION\_GROUP\_ALLOCATION\_ACTION flag of the Session-Group-Control-Vector AVP set.

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.

A Diameter server receiving a command for session initiation which includes at least one Session-Group-Info AVP but the server does not understand the semantics of this optional AVP because it does not support group operations according to the specification in this document, **MUST** ignore the optional group operations specific AVPs and proceed with processing the command for a single session.

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 with processing the command for a single session. Furthermore, the client keeps a log to remember that the server is not able to perform group operations.

#### 4.1.2. 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) to the client, indicating the session in the requests Session-Id AVP. The client sends a Re-Authorization Answer (RAA) 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.

#### 4.1.3. Mid-session group assignment modifications

Either Diameter peer 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 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 send 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.

#### 4.2. Session Grouping Capability Discovery

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

##### 4.2.1. 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 for capability exchange can be used to announce peers' support for session grouping and session group operations. In such case, the optional Session-Group-

Capability-Vector AVP, as described in Section 4.2.2 can be omitted in Diameter messages being exchanged between peers.

#### 4.2.2. Implicit Capability Discovery

If no explicit mechanism for capability discovery is deployed to enable Diameter nodes to learn about peers' capability to support session grouping and group commands, Diameter peers SHOULD append the Session-Group-Capability-Vector AVP to any Diameter messages exchanged with its peers to announce its capability to support session grouping and session group operations. Implementations following this specification 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 peer with the BASE\_SESSION\_GROUP\_CAPABILITY flag set, the Diameter node maintains a log to remember the peer's capability to support group commands.

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

#### 4.4.2. Receiving Group Commands

A Diameter peer 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.

The Diameter peer receiving a request which requests performing the command to at least on session group SHOULD perform follow up message exchanges according to the value identified in the Session-Group-Info AVP.

#### 4.4.3. Error Handling for Group Commands

When a Diameter peer 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 peer 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 Session 7.5 of [RFC6733].

#### 4.4.4. Single-Session Fallback

Either Diameter peer, a Diameter client or a Diameter server, 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. 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 Proxies Agents

This specification assumes in case of a present stateful Proxy Agent between a Diameter client and a Diameter server that the Proxy Agent is aware of session groups and session group handling. The Proxy MUST reflect the state of each session associated with a session group according to the result of a group command operated between a Diameter client and a server.

In case a Proxy Agent manipulates session groups, it MUST maintain consistency of session groups between a client and a server. This applies to deployment where the Proxy Agent utilizes session grouping and performing group commands with, for example, a Diameter server, whereas the Diameter client is not group-aware. The same applies to 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.

### 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 to perform the command to all sessions associated with the identified group(s).

### 6.1. Formatting Example: Group Re-Auth-Request

A request that one or more groups of users are re-authentication is issued by appending one or multiple Session-Group-Id AVP(s) to the Re-Auth-Request (RAR) and a single instance of a Group-Response-Action AVP. 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)

			+-----+   AVP Flag rules   +-----+			
Attribute Name	AVP Code	Value Type	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 element 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 peer 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

TODO

## 11. Acknowledgments

The authors of this document want to thank Ben Campbell and Eric McMurphy for their valuable comments to early versions of this draft.

## 12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4005] Calhoun, P., Zorn, G., Spence, D., and D. Mitton, "Diameter Network Access Server Application", RFC 4005, August 2005.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

## Appendix A. Session Management -- Exemplary Session State Machines

### A.1. 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 the additional state transitions related to the processing of the new 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.

A Diameter group command in the following tables is differentiated from a single-session related command by a preceding 'G'. 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].

State	Event	CLIENT, STATEFUL	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 client will comply with request to end the session	Send GASA with Result-Code = SUCCESS, Send STR per session	Discon
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 received.	Discon. user/device	Idle

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: September 10, 2020

M. Jones  
M. Liebsch  
L. Morand  
March 9, 2020

Diameter Group Signaling  
draft-ietf-dime-group-signaling-13.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 September 10, 2020.

Copyright Notice

Copyright (c) 2020 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 . . . . .	5
4. Protocol Description . . . . .	6
4.1. Session Grouping Capability Discovery . . . . .	6
4.1.1. Implicit Capability Discovery . . . . .	6
4.1.2. Explicit 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 . . . . .	10
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
9.2. New Registries . . . . .	20
10. Security Considerations . . . . .	20
11. Acknowledgments . . . . .	21
12. Normative References . . . . .	21

Appendix A. Session Management -- Exemplary Session State	
Machine . . . . .	22
A.1. Use of groups for the Authorization Session State Machine	22
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", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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 during an ongoing session (mid-session). For example, if a user's subscription profile changes mid-session, a Diameter server may remove a session from an existing group and assign this session to a different group that is more appropriate for the new subscription profile.

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

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. As example, the server issues a single Session Termination Request (STR) command, including the identifier of the group of sessions which are to be terminated. The Diameter client treats the STR as group command and initiates the termination of all sessions associated with the identified group. Subsequently, the client confirms the 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 client or a server decides to create a new session group, e.g., to group all sessions which share certain characteristics, this 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 restrict the permission to add or remove a session to/from a session group to the group owner. Either the client and the server can assign a session to a group. However, a session can be removed from a session group and/or moved to another session group only by the node that has assigned this session to the session group. A session group is deleted and its identifier released after the last session has been removed from this session group. The owner of a session group 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. A session group must only be deleted by the Diameter node that created it.

Diameter applications with implicit support for session groups MAY define a more constrained permission model. 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. Details about enforcing a more constraint permission model are out of scope of this specification.

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 the Diameter node did not create 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

#### 4. Protocol Description

##### 4.1. Session Grouping Capability Discovery

Diameter nodes SHOULD NOT perform group operations with peer nodes unless the node has advertised support for session grouping and group operations.

##### 4.1.1. Implicit Capability Discovery

Newly defined Diameter applications may natively support Diameter session grouping and group operations. Such applications provide intrinsic discovery for the support of session grouping capability using the assigned Application Id advertised during the capability

exchange phase two Diameter peers establish a transport connection (see Section 5.3 of [RFC6733]).

System- and deployment-specific means, as well as out-of-band mechanisms for capability discovery 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. Explicit Capability Discovery

If no other mechanism for capability discovery is deployed to enable Diameter nodes to learn about nodes' capability to support session grouping and group commands for a given application, a Diameter node SHOULD append the Session-Group-Capability-Vector AVP to any Diameter application messages exchanged with the other Diameter nodes to announce its capability to support session grouping and session group operations for the advertised application. Implementations following the specification as per this document MUST 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 receiving Diameter node discovers the supported session grouping capability of the sending Diameter node for the advertised application and MUST cache this information for the lifetime of the routing table entry associated with the peer identity/Application Id pair (see Section 2.7 of [RFC6733]).

#### 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 implementation of an application to determine such limitations. If 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 responsible for the session assignment to this group. 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 is 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 [RFC7155], 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. For all assignments of a session to an active session group made by the client or the server, the SESSION\_GROUP\_STATUS\_IND flag in the Session-Group-Info AVP, which identifies the session group, MUST be set. A set SESSION\_GROUP\_STATUS\_IND flag indicates that the identified session group has just been created or is still active.

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 includes 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. If one or multiple identified session groups are not already stored by the server, the server MUST store the newly 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 [RFC7155], 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 [RFC7155], 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 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 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 includes a Session-Group-Info AVP which does not include 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 includes 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. The Diameter client MUST 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 includes 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. If 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 or a service-specific request 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 or a service-specific answer. 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 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 Command Code Format (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 multiple resulting transactions associated with a group command are to be treated by appending a single instance of a Group-Response-Action AVP. For example, when a server sends a Re-Authorization Request (RAR) or a service-specific server-initiated request to the client, it indicates to the client to follow the request according to one of three possible procedures. When the server sets the Group-Response-Action AVP to ALL\_GROUPS (1), the client sends a single RAR message for all identified groups. When the server sets the Group-Response-Action AVP to PER\_GROUP (2), the client sends a single RAR message for each identified group individually. When the server sets the Group-Response-Action AVP to PER\_SESSION (3), the client follows-up with a single RAR message per impacted session. If a session is included in more than one of the identified session groups, the client sends only one RAR message for that session.



If the sender sends a request including the Group-Response-Action AVP set to ALL\_GROUPS (1) or PER\_GROUP (2), it has to 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 the 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 the case of a present stateful Proxy Agent between a Diameter client and a Diameter server, the Proxy Agent MUST perform the same mechanisms per this specification to advertise session grouping and group operations capability towards the client and the server respectively. 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.

If 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 are 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 control flags 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 include 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 unique. 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 document:

##### ALL\_GROUPS (1)

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

##### PER\_GROUP (2)

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

##### PER\_SESSION (3)

Follow up message exchanges associated with a group command should be performed with a separate 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.

### 9.2. New Registries

This specification requires IANA to create two registries:

- o Session-Group-Control-Vector AVP registry for control bits with two initial assignments, which are described in Section 7.2. The future registration assignment policy is proposed to be Specification Required.
- o Session-Group-Capability-Vector AVP with one initial assignment, which is described in Section 7.5. The future registration assignment policy is proposed to be Standards Action.

The AVP names can be used as registry names.

## 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. There is ongoing work on the specification of end-to-end security features for Diameter. Such features would 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 assumed 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 McMurphy 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>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.



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

## 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 client will comply with request to end the session	Send GASA with Result-Code = SUCCESS, Send STR per session	Discon
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	Send GRAA,	Pending

	Group-Response-Action = PER_SESSION, session is assigned to received group(s) and client will perform subsequent re-auth	Send service specific re-auth req per session	
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 received.	Discon. user/device	Idle

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-authoization 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: April 30, 2015

J. Korhonen, Ed.  
Broadcom  
S. Donovan, Ed.  
B. Campbell  
Oracle  
L. Morand  
Orange Labs  
October 27, 2014

Diameter Overload Indication Conveyance  
draft-ietf-dime-ovli-04.txt

Abstract

This specification documents a Diameter Overload Control (DOC) base solution and the dissemination of the overload report information.

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 April 30, 2015.

Copyright Notice

Copyright (c) 2014 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. Solution Overview . . . . .	5
3.1. Piggybacking Principle . . . . .	7
3.2. DOIC Capability Announcement . . . . .	8
3.3. DOIC Overload Condition Reporting . . . . .	9
3.4. DOIC Extensibility . . . . .	10
3.5. Simplified Example Architecture . . . . .	11
4. Solution Procedures . . . . .	12
4.1. Capability Announcement . . . . .	12
4.1.1. Reacting Node Behavior . . . . .	12
4.1.2. Reporting Node Behavior . . . . .	12
4.1.3. Agent Behavior . . . . .	13
4.2. Overload Report Processing . . . . .	14
4.2.1. Overload Control State . . . . .	14
4.2.2. Reacting Node Behavior . . . . .	18
4.2.3. Reporting Node Behavior . . . . .	18
4.3. Protocol Extensibility . . . . .	20
5. Loss Algorithm . . . . .	21
5.1. Overview . . . . .	21
5.2. Reporting Node Behavior . . . . .	22
5.3. Reacting Node Behavior . . . . .	22
6. Attribute Value Pairs . . . . .	23
6.1. OC-Supported-Features AVP . . . . .	23
6.2. OC-Feature-Vector AVP . . . . .	24
6.3. OC-OLR AVP . . . . .	24
6.4. OC-Sequence-Number AVP . . . . .	25
6.5. OC-Validity-Duration AVP . . . . .	25
6.6. OC-Report-Type AVP . . . . .	25
6.7. OC-Reduction-Percentage AVP . . . . .	26
6.8. Attribute Value Pair flag rules . . . . .	27
7. Error Response Codes . . . . .	27
8. IANA Considerations . . . . .	28
8.1. AVP codes . . . . .	28
8.2. New registries . . . . .	28
9. Security Considerations . . . . .	29
9.1. Potential Threat Modes . . . . .	29
9.2. Denial of Service Attacks . . . . .	30

9.3. Non-Compliant Nodes . . . . .	30
9.4. End-to End-Security Issues . . . . .	31
10. Contributors . . . . .	32
11. References . . . . .	32
11.1. Normative References . . . . .	32
11.2. Informative References . . . . .	32
Appendix A. Issues left for future specifications . . . . .	33
A.1. Additional traffic abatement algorithms . . . . .	33
A.2. Agent Overload . . . . .	33
A.3. New Error Diagnostic AVP . . . . .	33
Appendix B. Deployment Considerations . . . . .	34
Appendix C. Requirements Conformance Analysis . . . . .	34
Appendix D. Considerations for Applications Integrating the DOIC Solution . . . . .	34
D.1. Application Classification . . . . .	34
D.2. Application Type Overload Implications . . . . .	35
D.3. Request Transaction Classification . . . . .	36
D.4. Request Type Overload Implications . . . . .	37
Authors' Addresses . . . . .	38

## 1. Introduction

This specification defines a base solution for Diameter Overload Control (DOC), referred to as Diameter Overload Indication Conveyance (DOIC). The requirements for the solution are described and discussed in the corresponding design requirements document [RFC7068]. Note that the overload control solution defined in this specification does not address all the requirements listed in [RFC7068]. A number of overload control related features are left for the future specifications. See Appendix A for a list of extensions that are currently being considered. See Appendix C for an analysis of the conformance to the requirements specified in [RFC7068].

The solution defined in this specification addresses Diameter overload control between Diameter nodes that support the DOIC solution. Furthermore, the solution which is designed to apply to existing and future Diameter applications, requires no changes to the Diameter base protocol [RFC6733] and is deployable in environments where some Diameter nodes do not implement the Diameter overload control solution defined in this specification.

## 2. Terminology and Abbreviations

Abatement



Reaction to receipt of an overload report resulting in a reduction in traffic sent to the reporting node. Abatement actions include diversion and throttling.

#### Abatement Algorithm

An mechanism requested by reporting nodes and used by reacting nodes to reduce the amount of traffic sent during an occurrence of overload control.

#### Diversion

Abatement of traffic sent to a reporting node by a reacting node in response to receipt of an overload report. The abatement is achieved by diverting traffic from the reporting node to another Diameter node that is able to process the request.

#### Host-Routed Request

The set of requests that a reacting node knows will be served by a particular host, either due to the presence of a Destination-Host AVP, or by some other local knowledge on the part of the reacting node.

#### Overload Control State (OCS)

Reporting and reacting node internally maintained state describing occurrences of overload control.

#### Overload Report (OLR)

Information sent by a reporting node indicating the start, continuation or end of an occurrence of overload control.

#### Reacting Node

A Diameter node that acts upon an overload report.

#### Realm-Routed Request

The set of requests that a reacting node does not know the host that will service the request.

#### Reporting Node

A Diameter node that generates an overload report. (This may or may not be the overloaded node.)

## Throttling

Throttling is the reduction of the number of requests sent to an entity. Throttling can include a Diameter Client or Diameter Server dropping requests, or a Diameter Agent rejecting requests with appropriate error responses. In extreme cases reporting nodes can also throttle requests when the requested reductions in traffic does not sufficiently address the overload scenario.

## 3. Solution Overview

The Diameter Overload Information Conveyance (DOIC) solution allows Diameter nodes to request other nodes to perform overload abatement actions, that is, actions to reduce the load offered to the overloaded node or realm.

A Diameter node that supports DOIC is known as a "DOIC node". Any Diameter node can act as a DOIC node, including clients, servers, and agents. DOIC nodes are further divided into "Reporting Nodes" and "Reacting Nodes." A reporting node requests overload abatement by sending an Overload Report (OLR) to one or more reacting nodes.

A reacting node acts upon OLRs, and performs whatever actions are needed to fulfil the abatement requests included in the OLRs. A Reporting node may report overload on its own behalf, or on behalf of other (typically upstream) nodes. Likewise, a reacting node may perform overload abatement on its own behalf, or on behalf of other (typically downstream) nodes.

A node's role as a DOIC node is independent of its Diameter role. For example, Diameter Relay and Proxy Agents may act as DOIC nodes, even though they are not endpoints in the Diameter sense. Since Diameter enables bi-directional applications, where Diameter Servers can send requests towards Diameter Clients, a given Diameter node can simultaneously act as a reporting node and a reacting node.

Likewise, a relay or proxy agent may act as a reacting node from the perspective of upstream nodes, and a reporting node from the perspective of downstream nodes.

DOIC nodes do not generate new messages to carry DOIC related information. Rather, they "piggyback" DOIC information over existing Diameter messages by inserting new AVPs into existing Diameter requests and responses. Nodes indicate support for DOIC, and any needed DOIC parameters by inserting an OC\_Supported\_Features AVP

(Section 6.2) into existing requests and responses. Reporting nodes send OLRs by inserting OC-OLR AVPs (Section 6.3).

A given OLR applies to the Diameter realm and application of the Diameter message that carries it. If a reporting node supports more than one realm and/or application, it reports independently for each combination of realm and application. Similarly, the OC-Supported-Features AVP applies to the realm and application of the enclosing message. This implies that a node may support DOIC for one application and/or realm, but not another, and may indicate different DOIC parameters for each application and realm for which it supports DOIC.

Reacting nodes perform overload abatement according to an agreed-upon abatement algorithm. An abatement algorithm defines the meaning of the parameters of an OLR and the procedures required for overload abatement. This document specifies a single must-support algorithm, namely the "loss" algorithm (Section 5). Future specifications may introduce new algorithms.

Overload conditions may vary in scope. For example, a single Diameter node may be overloaded, in which case reacting nodes may reasonably attempt to send requests to other destinations or via other agents. On the other hand, an entire Diameter realm may be overloaded, in which case such attempts would do harm. DOIC OLRs have a concept of "report type" (Section 6.6), where the type defines such behaviors. Report types are extensible. This document defines report types for overload of a specific server, and for overload of an entire realm.

A report of type host is sent to indicate the overload of a specific server for the application-id indicated in the transaction. When receiving an OLR of type host, a reacting node applies overload abatement to what is referred to in this document as host-routed requests. This is the set of requests that the reacting node knows will be served by a particular host, either due to the presence of a Destination-Host AVP, or by some other local knowledge on the part of the reacting node. The reacting node applies overload abatement on those host-routed requests which the reacting node knows will be served by the server that matches the Origin-Host AVP of the received message that contained the received OLR of type host.

A report type of realm is sent to indicate the overload of all servers in a realm for the application-id. When receiving an OLR of type realm, a reacting node applies overload abatement to what is referred to in this document as realm-routed requests. This is the set of requests that are not host-routed as defined in the previous paragraph.

While a reporting node sends OLRs to "adjacent" reacting nodes, nodes that are "adjacent" for DOIC purposes may not be adjacent from a Diameter, or transport, perspective. For example, one or more Diameter agents that do not support DOIC may exist between a given pair of reporting and reacting nodes, as long as those agents pass unknown AVPs through unchanged. The report types described in this document can safely pass through non-supporting agents. This may not be true for report types defined in future specifications. Documents that introduce new report types MUST describe any limitations on their use across non-supporting agents.

### 3.1. Piggybacking Principle

The overload control AVPs defined in this specification have been designed to be piggybacked on top of existing application messages. This is made possible by adding overload control top-level AVPs, the OC-OLR AVP and the OC-Supported-Features AVP, as optional AVPs into existing commands when the corresponding Command Code Format (CCF) specification allows adding new optional AVPs (see Section 1.3.4 of [RFC6733]).

Reacting nodes indicate support for DOIC by including the OC-Supported-Features AVP in all request messages originated or relayed by the reacting node.

Reporting nodes indicate support for DOIC by including the OC-Supported-Features AVP in all answer messages originated or relayed by the reporting node. Reporting nodes also include overload reports using the OC-OLR AVP in answer messages.

Note: There is no new Diameter application defined to carry overload related AVPs. The DOIC AVPs are carried in existing Diameter application messages.

Note that the overload control solution does not have fixed server and client roles. The DOIC node role is determined based on the message type: whether the message is a request (i.e. sent by a "reacting node") or an answer (i.e. sent by a "reporting node"). Therefore, in a typical "client-server" deployment, the Diameter Client MAY report its overload condition to the Diameter Server for any Diameter Server initiated message exchange. An example of such is the Diameter Server requesting a re-authentication from a Diameter Client.

### 3.2. DOIC Capability Announcement

The DOIC solution supports the ability for Diameter nodes to determine if other nodes in the path of a request support the solution. This capability is referred to as DOIC Capability Announcement (DCA) and is separate from Diameter Capability Exchange.

The DCA solution uses the OC-Supported-Features AVPs to indicate the Diameter overload features supported.

The first node in the path of a Diameter request that supports the DOIC solution inserts the OC-Supported-Feature AVP in the request message. This includes an indication that it supports the loss overload abatement algorithm defined in this specification (see Section 5). This ensures that there is at least one commonly supported overload abatement algorithm between the reporting node and the reacting nodes in the path of the request.

DOIC must support deployments where Diameter Clients and/or Diameter Servers do not support the DOIC solution. In this scenario, it is assumed that Diameter Agents that support the DOIC solution will handle overload abatement for the non supporting Diameter nodes. In this case the DOIC agent will insert the OC-Supported-Features AVP in requests that do not already contain one, telling the reporting node that there is a DOIC node that will handle overload abatement.

The reporting node inserts the OC-Supported-Feature AVP in all answer messages to requests that contained the OC-Supported-Feature AVP. The contents of the reporting node's OC-Supported-Feature AVP indicate the set of Diameter overload features supported by the reporting node with one exception.

The reporting node only includes an indication of support for one overload abatement algorithm. This is the algorithm that the reporting node intends to use should it enter an overload condition or requests to use while it actually is in an overload condition. Reacting nodes can use the indicated overload abatement algorithm to prepare for possible overload reports and must use the indicated overload abatement algorithm if traffic reduction is actually requested.

Note that the loss algorithm defined in this document is a stateless abatement algorithm. As a result it does not require any actions by reacting nodes prior to the receipt of an overload report. Stateful abatement algorithms that base the abatement logic on a history of request messages sent might require reacting

nodes to maintain state to ensure that overload reports can be properly handled.

The individual features supported by the DOIC nodes are indicated in the OC-Feature-Vector AVP. Any semantics associated with the features will be defined in extension specifications that introduce the features.

The DCA mechanism must also support the scenario where the set of features supported by the sender of a request and by agents in the path of a request differ. In this case, the agent updates the OC-Supported-Feature AVP to reflect the mixture of the two sets of supported features.

The logic to determine the content of the modified OC-Supported-Feature AVP is out-of-scope for this specification and is left to implementation decisions. Care must be taken not to introduce interoperability issues for downstream or upstream DOIC nodes.

### 3.3. DOIC Overload Condition Reporting

As with DOIC Capability Announcement, Overload Condition Reporting uses new AVPs (Section 6.3) to indicate an overload condition.

The OC-OLR AVP is referred to as an overload report. The OC-OLR AVP includes the type of report, a sequence number, the length of time that the report is valid and abatement algorithm specific AVPs.

Two types of overload reports are defined in this document, host reports and realm reports.

A report of type host is sent to indicate the overload of a specific Diameter node for the application-id indicated in the transaction. When receiving an OLR of type host, a reacting node applies overload abatement to what is referred to in this document as host-routed requests. This is the set of requests that the reacting node knows will be served by a particular host, either due to the presence of a Destination-Host AVP, or by some other local knowledge on the part of the reacting node. The reacting node applies overload abatement on those host-routed requests which the reacting node knows will be served by the server that matches the Origin-Host AVP of the received message that contained the received OLR of type host.

Realm reports apply to realm-routed requests for a specific realm as indicated in the Destination-Realm AVP.

Reporting nodes are responsible for determining the need for a reduction of traffic. The method for making this determination is

implementation specific and depend on the type of overload report being generated. A host report, for instance, will generally be generated by tracking utilization of resources required by the host to handle transactions for the Diameter application. A realm report will generally impact the traffic sent to multiple hosts and, as such, will typically require tracking the capacity of the servers able to handle realm-routed requests for the application.

Once a reporting node determines the need for a reduction in traffic, it uses the DOIC defined AVPs to report on the condition. These AVPs are included in answer messages sent or relayed by the reporting node. The reporting node indicates the overload abatement algorithm that is to be used to handle the traffic reduction in the OC-Supported-Features AVP. The OC-OLR AVP is used to communicate information about the requested reduction.

Reacting nodes, upon receipt of an overload report, are responsible for applying the abatement algorithm to traffic impacted by the overload report. The method used for that abatement is dependent on the abatement algorithm. The loss abatement algorithm is defined in this document (Section 5). Other abatement algorithms can be defined in extensions to the DOIC solutions.

As the conditions that lead to the generation of the overload report change the reporting node can send new overload reports requesting greater reduction if the condition gets worse or less reduction if the condition improves. The reporting node sends an overload report with a duration of zero to indicate that the overload condition has ended and use of the abatement algorithm is no longer needed.

The reacting node also determines when the overload report expires based on the OC-Validity-Duration AVP in the overload report and stops applying the abatement algorithm when the report expires.

### 3.4. DOIC Extensibility

The DOIC solution is designed to be extensible. This extensibility is based on existing Diameter based extensibility mechanisms.

There are multiple categories of extensions that are expected. This includes the definition of new overload abatement algorithms, the definition of new report types and new definitions of the scope of messages impacted by an overload report.

The DOIC solution uses the OC-Supported-Features AVP for DOIC nodes to communicate supported features. The specific features supported by the DOIC node are indicated in the OC-Feature-Vector AVP. DOIC extensions must define new values for the OC-Feature-Vector AVP.

DOIC extensions also have the ability to add new AVPs to the OC-Supported-Features AVP, if additional information about the new feature is required.

Reporting nodes use the OC-OLR AVP to communicate overload occurrences. This AVP can also be extended to add new AVPs allowing a reporting nodes to communicate additional information about handling an overload condition.

If necessary, new extensions can also define new top-level AVPs. It is, however, recommended that DOIC extensions use the OC-Supported-Features and OC-OLR to carry all DOIC related AVPs.

### 3.5. Simplified Example Architecture

Figure 1 illustrates the simplified architecture for Diameter overload information conveyance.

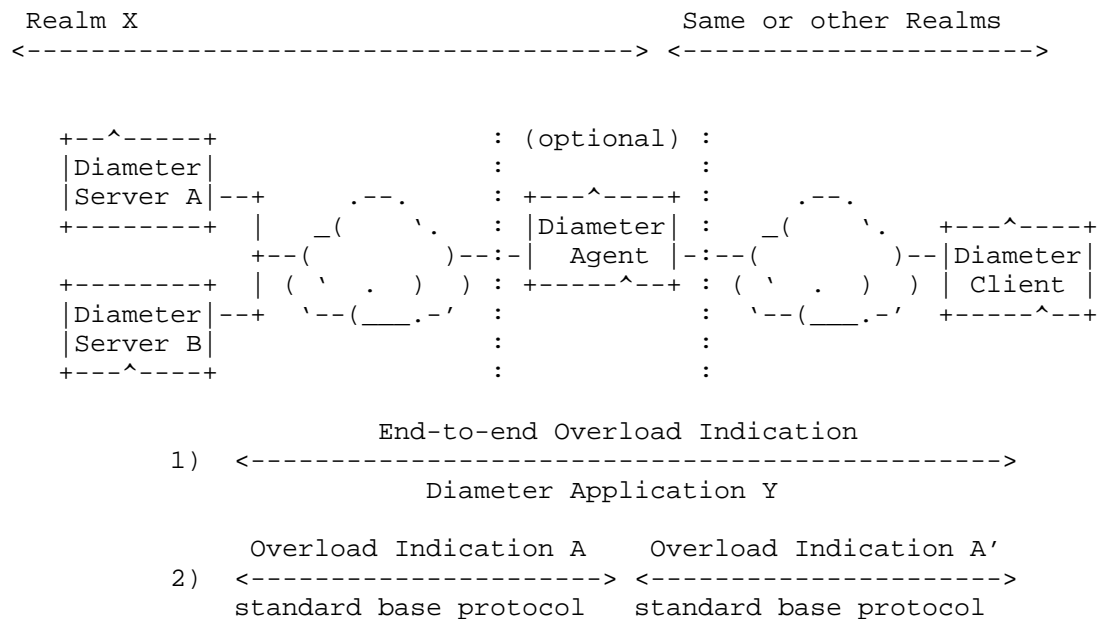


Figure 1: Simplified architecture choices for overload indication delivery

In Figure 1, the Diameter overload indication can be conveyed (1) end-to-end between servers and clients or (2) between servers and



Diameter agent inside the realm and then between the Diameter agent and the clients.

#### 4. Solution Procedures

This section outlines the normative behavior associated with the DOIC solution.

##### 4.1. Capability Announcement

This section defines DOIC Capability Announcement (DCA) behavior.

###### 4.1.1. Reacting Node Behavior

A reacting node **MUST** include the OC-Supported-Features AVP in all request messages.

A reacting node **MAY** include the OC-Feature-Vector AVP with an indication of the loss algorithm. A reacting node **MUST** include the OC-Feature-Vector AVP to indicate support for abatement algorithms in addition to the loss algorithm.

A reacting node **SHOULD** indicate support for all other DOIC features it supports.

Not all DOIC features will necessarily apply to all transactions. For instance, there may be a future extension that only applies to session based applications. A reacting node that supports this extension can choose to not include it for non session based applications.

An OC-Supported-Features AVP in answer messages indicates there is a reporting node for the transaction. The reacting node **MAY** take action based on the features indicated in the OC-Feature-Vector AVP.

Note that the loss abatement algorithm is the only feature described in this document and it does not require action to be taken when there is an active overload report. This behavior is described in Section 4.2 and Section 5.

###### 4.1.2. Reporting Node Behavior

Upon receipt of a request message, a reporting node determines if there is a reacting node for the transaction based on the presence of the OC-Supported-Features AVP.

If the request message contains an OC-Supported-Features AVP then the reporting node MUST include the OC-Supported-Features AVP in the answer message for that transaction.

The reporting node MUST NOT include the OC-Supported-Features AVP, OC-OLR AVP or any other overload control AVPs defined in extension drafts in response messages for transactions where the request message does not include the OC-Supported-Features AVP. Lack of the OC-Supported-Features AVP in the request message indicates that there is no reacting node for the transaction.

Based on the content of the OC-Supported-Features AVP in the request message, the reporting node knows what overload control functionality is supported by the reacting node. The reporting node then acts accordingly for the subsequent answer messages it initiates.

The reporting node MUST indicate support for one and only one abatement algorithm in the OC-Feature-Vector AVP. The abatement algorithm included MUST be from the set of abatement algorithms contained in the request message's OC-Supported-Features AVP. The abatement algorithm included MUST indicate the abatement algorithm the reporting node wants the reacting node to use when the reporting node enters an overload condition.

For an ongoing overload state, a reacting node MUST keep the algorithm that was selected by the reporting node in further requests towards the reporting node. The reporting node SHOULD NOT change the selected algorithm during a period of time that it is in an overload condition and, as a result, is sending OC-OLR AVPs in answer messages.

The reporting node SHOULD indicate support for other DOIC features defined in extension drafts that it supports and that apply to the transaction.

Note that not all DOIC features will apply to all Diameter applications or deployment scenarios. The features included in the OC-Feature-Vector AVP are based on local reporting node policy.

#### 4.1.3. Agent Behavior

Diameter agents that support DOIC MUST ensure that all messages have the OC-Supporting-Features AVP. If a message handled by the DOIC agent does not include the OC-Supported-Features AVP then the DOIC agent inserts the AVP. If the message already has the AVP then the agent either leaves it unchanged in the relayed message or modifies it to reflect a mixed set of DOIC features.

An agent MAY modify the OC-Supported-Features AVP carried in answer messages.

For instance, if the agent supports a superset of the features reported by the reacting node then the agent might choose, based on local policy, to advertise that superset of features to the reporting node.

If the agent modifies the OC-Supported-Features AVP sent to the reporting node then it might also need to modify the OC-Supported-Features AVP sent to a reacting node in the subsequent answer message, as it cannot send an indication of support for features that are not supported by the reacting node.

Editor's note: There is an open issue on the wording around agent behavior in this case that needs to be resolved prior to finishing this document.

## 4.2. Overload Report Processing

### 4.2.1. Overload Control State

Both reacting and reporting nodes maintain Overload Control State (OCS) for active overload conditions.

#### 4.2.1.1. Overload Control State for Reacting Nodes

A reacting node SHOULD maintain the following OCS per supported Diameter application:

- o A host-type OCS entry for each Destination-Host to which it sends host-type requests and
- o A realm-type OCS entry for each Destination-Realm to which it sends realm-type requests.

A host-type OCS entry is identified by the pair of Application-Id and Host-Id.

A realm-type OCS entry is identified by the pair of Application-Id and Realm-Id.

The host-type and realm-type OCS entries MAY include the following information (the actual information stored is an implementation decision):

- o Sequence number (as received in OC-OLR)

- o 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)
- o Selected Abatement Algorithm (as received in OC-Supported-Features AVP)
- o Abatement Algorithm specific input data (as received within the OC-OLR AVP, for example, OC-Reduction-Percentage for the Loss abatement algorithm)

#### 4.2.1.2. Overload Control State for Reporting Nodes

A reporting node SHOULD maintain OCS entries per supported Diameter application, per supported (and eventually selected) Abatement Algorithm and per report-type.

An OCS entry is identified by the pair of Application-Id and Abatement Algorithm.

The OCS entry for a given pair of Application and Abatement Algorithm MAY include the information (the actual information stored is an implementation decision):

- o Report type
- o Sequence number
- o Validity Duration
- o Expiration Time
- o Algorithm specific input data (for example, the Reduction Percentage for the Loss Abatement Algorithm)

#### 4.2.1.3. Reacting Node Maintenance of Overload Control State

When a reacting node receives an OC-OLR AVP, it MUST determine if it is for an existing or new overload condition.

For the remainder of this section the term OLR refers to the combination of the contents of the received OC-OLR AVP and the abatement algorithm indicated in the received OC-Supported-Features AVP.

The OLR is for an existing overload condition if the reacting node has an OCS that matches the received OLR.

For a host report-type this means it matches the app-id and host-id in an existing host OCS entry.

For a realm report-type this means it matches the app-id and realm-id in an existing realm OCS entry.

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

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

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

For a host report-type this means it creates an OCS entry with the app-id of the application-id in the received message and host-id of the Origin-Host in the received message.

Note: This solution assumes that the Origin-Host AVP in the answer message included by the reporting node is not changed along the path to the reacting node.

For a realm report-type this means it creates an OCS entry with the app-id of the application-id in the received message and realm-id of the Origin-Realm in the received message.

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

Note that it is not necessarily appropriate to delete the OCS entry, as there is recommended behavior that the reacting node slowly returns to full traffic when ending an overload abatement period.

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").

#### 4.2.1.4. Reporting Node Maintenance of Overload Control State

A reporting node SHOULD create a new OCS entry when entering an overload condition.

If the reporting node knows through absence of the OC-Supported-Features AVP in received messages that there are no reacting nodes supporting DOIC then the reporting node can choose to not create OCS entries.

When generating a new OCS entry the sequence number MAY be set to any value if there is no unexpired overload report for previous overload conditions sent to any reacting node for the same application and report-type.

When generating sequence numbers for new overload conditions, the new sequence number MUST be greater than any sequence number in an active (unexpired) overload report previously sent by the reporting node. This property MUST hold over a reboot of the reporting node.

The reporting node MUST update an OCS entry when it needs to adjust the validity duration of the overload condition at reacting nodes.

For instance, if the reporting node wishes to instruct reacting nodes to continue overload abatement for a longer period of time that originally communicated. This also applies if the reporting node wishes to shorten the period of time that overload abatement is to continue.

A reporting node MUST NOT update the abatement algorithm in an active OCS entry.

A reporting node MUST update an OCS entry when it wishes to adjust any abatement algorithm specific parameters, including the reduction percentage used for the Loss abatement algorithm.

For instance, if the reporting node wishes to change the reduction percentage either higher, if the overload condition has worsened, or lower, if the overload condition has improved, then the reporting node would update the appropriate OCS entry.

The reporting node MUST update the sequence number associated with the OCS entry anytime the contents of the OCS entry are changed. This will result in a new sequence number being sent to reacting nodes, instructing the reacting nodes to process the OC-OLR AVP.

A reporting node SHOULD update an OCS entry with a validity duration of zero ("0") when the overload condition ends.

If the reporting node knows that the OCS entries in the reacting nodes are near expiration then the reporting node can decide to delete the OCS entry.

The reporting node MUST keep an OCS entry with a validity duration of zero ("0") for a period of time long enough to ensure that any non-expired reacting node's OCS entry created as a result of the overload condition in the reporting node is deleted.

#### 4.2.2. Reacting Node Behavior

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

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

For the Loss abatement algorithm defined in this specification, see Section 5 for the abatement logic applied.

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 as defined in section Section 7.

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

#### 4.2.3. Reporting Node Behavior

The operation on the reporting node is straight forward.

If there is an active OCS entry then the reporting node SHOULD include the OC-OLR AVP in all answer messages to requests that contain the OC-Supported-Features AVP and that match the active OCS entry.

A request matches if the application-id in the request matches the application-id in any active OCS entry and if the report-type in the OCS entry matches a report-type supported by the reporting node as indicated in the OC-Supported-Features AVP.

The contents of the OC-OLR AVP MUST contain all information necessary for the abatement algorithm indicated in the OC-Supported-Features AVP that is also included in the answer message.

A reporting node MAY choose to not resend an overload report to a reacting node if it can guarantee that this overload report is already active in the reacting node.

Note - In some cases (e.g. when there are one or more agents in the path between reporting and reacting nodes, or when overload reports are discarded by reacting nodes) the reporting node may not be able to guarantee that the reacting node has received the report.

A reporting node MUST NOT send overload reports of a type that has not been advertised as supported by the reacting node.

Note that a reacting node advertises support for the host and realm report types by including the OC-Supported-Features AVP in the request. Support for other report types must be explicitly indicated by new feature bits in the OC-Feature-Vector AVP.

A reporting node MAY rely on the OC-Validity-Duration AVP values for the implicit overload control state cleanup on the reacting node. However, it is RECOMMENDED that the reporting node always explicitly indicates the end of a overload condition.

The reporting node SHOULD indicate the end of an overload occurrence by sending a new OLR with OC-Validity-Duration set to a value of zero ("0"). The reporting node SHOULD ensure that all reacting nodes receive the updated overload report.

All OLRs sent have an expiration time calculated by adding the validity-duration contained in the OLR to the time the message was sent. Transit time for the OLR can be safely ignored. The reporting node can ensure that all reacting nodes have received the OLR by continuing to send it in answer messages until the expiration time for all OLRs sent for that overload condition have expired.

When a reporting node sends an OLR, it effectively delegates any necessary throttling to downstream nodes. Therefore, the reporting node SHOULD NOT apply throttling to the set of messages to which the OLR applies. That is, the same candidate set of messages SHOULD NOT be throttled multiple times.

However, when the reporting node sends an OLR downstream, it MAY still be responsible to apply other abatement methods such as diversion. The reporting node might also need to throttle requests for reasons other than overload. For example, an agent or server might have a configured rate limit for each client, and throttle



requests that exceed that limit, even if such requests had already been candidates for throttling by downstream nodes.

This document assumes that there is a single source for realm-reports for a given realm, or that if multiple nodes can send realm reports, that each such node has full knowledge of the overload state of the entire realm. A reacting node cannot distinguish between receiving realm-reports from a single node, or from multiple nodes.

Editor's Note: There is not yet consensus on the above two paragraphs. Two alternatives are under consideration -- synchronization of sequence numbers and attribution of reports. If no consensus is reached then it will be left to be addressed as an extension.

#### 4.3. Protocol Extensibility

The overload control solution can be extended, e.g. with new traffic abatement algorithms, new report types or other new functionality.

When defining a new extension a new feature bit MUST be defined for the OC-Feature-Vector. This feature bit is used to communicate support for the new feature.

The extension MAY define new AVPs for use in DOIC Capability Announcement and for use in DOIC Overload reporting. These new AVPs SHOULD be defined to be extensions to the OC-Supported-Features and OC-OLR AVPs defined in this document.

It should be noted that [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.

The handling of feature bits in the OC-Feature-Vector AVP that are not associated with overload abatement algorithms MUST be specified by the extensions that define the features.

When defining new report type values, the corresponding specification MUST define the semantics of the new report types and how they affect the OC-OLR AVP handling. The specification MUST also reserve a corresponding new feature bit in the OC-Feature-Vector AVP.

The OC-OLR AVP can be expanded with optional sub-AVPs only if a legacy DOIC implementation can safely ignore them without breaking backward compatibility for the given OC-Report-Type AVP value. If the new sub-AVPs imply new semantics for handling the indicated report type, then a new OC-Report-Type AVP value MUST be defined.

New features (feature bits in the OC-Feature-Vector AVP) and report types (in the OC-Report-Type AVP) MUST be registered with IANA. As with any Diameter specification, new AVPs MUST also be registered with IANA. See Section 8 for the required procedures.

## 5. Loss Algorithm

This section documents the Diameter overload loss abatement algorithm.

### 5.1. Overview

The DOIC specification supports the ability for multiple overload abatement algorithms to be specified. The abatement algorithm used for any instance of overload is determined by the Diameter Overload Capability Announcement process documented in Section 4.1.

The loss algorithm described in this section is the default algorithm that must be supported by all Diameter nodes that support DOIC.

The loss algorithm is designed to be a straightforward and stateless overload abatement algorithm. It is used by reporting nodes to request a percentage reduction in the amount of traffic sent. The traffic impacted by the requested reduction depends on the type of overload report.

Reporting nodes use a strategy of applying abatement logic to the requested percentage of request messages sent (or handled in the case of agents) by the reacting node that are impacted by the overload report.

From a conceptual level, the logic at the reacting node could be outlined as follows.

1. An overload report is received and the associated overload state is either saved or updated (if required) by the reacting node.
2. A new Diameter request is generated by the application running on the reacting node.
3. The reacting node determines that an active overload report applies to the request, as indicated by the corresponding OCS entry.
4. The reacting node determines if abatement should be applied to the request. One approach that could be taken for each request is to select a random number between 1 and 100. If the random number is less than the indicated reduction percentage then the

request is given abatement treatment, otherwise the request is given normal routing treatment.

## 5.2. Reporting Node Behavior

The method a reporting nodes uses to determine the amount of traffic reduction required to address an overload condition is an implementation decision.

When a reporting node that has selected the loss abatement algorithm determines the need to request a traffic reduction it includes an OC-OLR AVP in response messages as described in Section 4.2.3.

The reporting node **MUST** indicate a percentage reduction in the OC-Reduction-Percentage AVP.

The reporting node **MAY** change the reduction percentage in subsequent overload reports. When doing so the reporting node must conform to overload report handling specified in Section 4.2.3.

When the reporting node determines it no longer needs a reduction in traffic the reporting node **SHOULD** send an overload report indicating the overload report is no longer valid, as specified in Section 4.2.3.

## 5.3. Reacting Node Behavior

The method a reacting node uses to determine which request messages are given abatement treatment is an implementation decision.

When receiving an OC-OLR in an answer message where the algorithm indicated in the OC-Supported-Features AVP is the loss algorithm, the reacting node **MUST** apply abatement treatment to the requested percentage of request messages sent.

Note: the loss algorithm is a stateless algorithm. As a result, the reacting node does not guarantee that there will be an absolute reduction in traffic sent. Rather, it guarantees that the requested percentage of new requests will be given abatement treatment.

When applying overload abatement treatment for the load abatement algorithm, the reacting node **MUST** abate, either by throttling or diversion, the requested percentage of requests that would have otherwise been sent to the reporting host or realm.

If reacting node comes out of the 100 percent traffic reduction as a result of the overload report timing out, the following concerns are

RECOMMENDED to be applied. The reacting node sending the traffic should be conservative and, for example, first send "probe" messages to learn the overload condition of the overloaded node before converging to any traffic amount/rate decided by the sender. Similar concerns apply in all cases when the overload report times out unless the previous overload report stated 0 percent reduction.

If the reacting node does not receive an OLR in messages sent to the formerly overloaded node then the reacting node SHOULD slowly increase the rate of traffic sent to the overloaded node.

It is suggested that the reacting node decrease the amount of traffic given abatement treatment by 20% each second until the reduction is completely removed and no traffic is given abatement treatment.

The goal of this behavior is to reduce the probability of overload condition thrashing where an immediate transition from 100% reduction to 0% reduction results in the reporting node moving quickly back into an overload condition.

## 6. Attribute Value Pairs

This section describes the encoding and semantics of the Diameter Overload Indication Attribute Value Pairs (AVPs) defined in this document.

A new application specification can incorporate the overload control mechanism specified in this document by making it mandatory to implement for the application and referencing this specification normatively. It is the responsibility of the Diameter application designers to define how overload control mechanisms works on that application.

### 6.1. OC-Supported-Features AVP

The OC-Supported-Features AVP (AVP code TBD1) is type of Grouped and serves two purposes. First, it announces a node's support for the DOIC solution in general. Second, it contains the description of the supported DOIC features of the sending node. The OC-Supported-Features AVP MUST be included in every Diameter request message a DOIC supporting node sends.

```
OC-Supported-Features ::= < AVP Header: TBD1 >
                        [ OC-Feature-Vector ]
                        * [ AVP ]
```

The OC-Feature-Vector sub-AVP is used to announce the DOIC features supported by the DOIC node, in the form of a flag bits field in which each bit announces one feature or capability supported by the node (see Section 6.2). The absence of the OC-Feature-Vector AVP indicates that only the default traffic abatement algorithm described in this specification is supported.

## 6.2. OC-Feature-Vector AVP

The OC-Feature-Vector AVP (AVP code TBD6) is type of Unsigned64 and contains a 64 bit flags field of announced capabilities of a DOIC node. The value of zero (0) is reserved.

The following capabilities are defined in this document:

OLR\_DEFAULT\_ALGO (0x0000000000000001)

When this flag is set by the DOIC node it means that the default traffic abatement (loss) algorithm is supported.

## 6.3. OC-OLR AVP

The OC-OLR AVP (AVP code TBD2) is type of Grouped and contains the information necessary to convey an overload report on an overload condition at the reporting node. The OC-OLR AVP does not explicitly contain all information needed by the reacting node to decide whether a subsequent request must undergo a throttling process with the received reduction percentage. The value of the OC-Report-Type AVP within the OC-OLR AVP indicates which implicit information is relevant for this decision (see Section 6.6). The application the OC-OLR AVP applies to is the same as the Application-Id found in the Diameter message header. The host or realm the OC-OLR AVP concerns is determined from the Origin-Host AVP and/or Origin-Realm AVP found in the encapsulating Diameter command. The OC-OLR AVP is intended to be sent only by a reporting node.

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

Note that if a Diameter command were to contain multiple OC-OLR AVPs they all MUST have different OC-Report-Type AVP value. OC-OLR AVPs with unknown values SHOULD be silently discarded by reacting nodes and the event SHOULD be logged.

#### 6.4. OC-Sequence-Number AVP

The OC-Sequence-Number AVP (AVP code TBD3) is type of Unsigned64. Its usage in the context of overload control is described in Section 4.2.

From the functionality point of view, the OC-Sequence-Number AVP MUST be used as a non-volatile increasing counter for a sequence of overload reports between two DOIC nodes for the same overload occurrence. The sequence number is only required to be unique between two DOIC nodes. Sequence numbers are treated in a uni-directional manner, i.e. two sequence numbers on each direction between two DOIC nodes are not related or correlated.

#### 6.5. OC-Validity-Duration AVP

The OC-Validity-Duration AVP (AVP code TBD4) is type of Unsigned32 and indicates in milliseconds the validity time of the overload report. The number of milliseconds is measured after reception of the first OC-OLR AVP with a given value of OC-Sequence-Number AVP. The default value for the OC-Validity-Duration AVP is 5000 (i.e., 5 seconds). When the OC-Validity-Duration AVP is not present in the OC-OLR AVP, the default value applies. Validity duration with values above 86400 (i.e.; 24 hours) MUST NOT be used. Invalid duration values are treated as if the OC-Validity-Duration AVP were not present and result in the default value being used.

Editor's note: There is an open discussion on whether to have an upper limit on the OC-Validity-Duration value, beyond that which can be indicated by an Unsigned32.

A timeout of the overload report has specific concerns that need to be taken into account by the DOIC node acting on the earlier received overload report(s). Section 6.7 discusses the impacts of timeout in the scope of the traffic abatement algorithms.

#### 6.6. OC-Report-Type AVP

The OC-Report-Type AVP (AVP code TBD5) is type of Enumerated. The value of the AVP describes what the overload report concerns. The following values are initially defined:

- 0 A host report. The overload treatment should apply to requests for which all of the following conditions are true:

Either the Destination-Host AVP is present in the request and its value matches the value of the Origin-Host AVP of the received message that contained the OC-OLR AVP; or the Destination-Host is

not present in the request but the value of the peer identity associated with the connection used to send the request matches the value of the Origin-Host AVP of the received message that contained the OC-OLR AVP.

The value of the Destination-Realm AVP in the request matches the value of the Origin-Realm AVP of the received message that contained the OC-OLR AVP.

The value of the Application-ID in the Diameter Header of the request matches the value of the Application-ID of the Diameter Header of the received message that contained the OC-OLR AVP.

- 1 A realm report. The overload treatment should apply to requests for which all of the following conditions are true:

The Destination-Host AVP is absent in the request and the value of the peer identity associated with the connection used to send the request does not match a server that could serve the request.

The value of the Destination-Realm AVP in the request matches the value of the Origin-Realm AVP of the received message that contained the OC-OLR AVP.

The value of the Application-ID in the Diameter Header of the request matches the value of the Application-ID of the Diameter Header of the received message that contained the OC-OLR AVP.

The OC-Report-Type AVP is envisioned to be useful for situations where a reacting node needs to apply different overload treatments for different overload contexts. For example, the reacting node(s) might need to throttle differently requests sent to a specific server (identified by the Destination-Host AVP in the request) and requests that can be handled by any server in a realm.

#### 6.7. OC-Reduction-Percentage AVP

The OC-Reduction-Percentage AVP (AVP code TBD8) is type of Unsigned32 and describes the percentage of the traffic that the sender is requested to reduce, compared to what it otherwise would send. The OC-Reduction-Percentage AVP applies to the default (loss) algorithm specified in this specification. However, the AVP can be reused for future abatement algorithms, if its semantics fit into the new algorithm.

The value of the Reduction-Percentage AVP is between zero (0) and one hundred (100). Values greater than 100 are ignored. The value of 100 means that all traffic is to be throttled, i.e. the reporting

node is under a severe load and ceases to process any new messages. The value of 0 means that the reporting node is in a stable state and has no need for the reacting node to apply any traffic abatement. The default value of the OC-Reduction-Percentage AVP is 0. When the OC-Reduction-Percentage AVP is not present in the overload report, the default value applies.

#### 6.8. Attribute Value Pair flag rules

				+-----+	
				AVP flag	
				rules	
				+-----+	+-----+
Attribute Name	AVP Code	Section Defined	Value Type	MUST	MUST
				NOT	NOT
OC-Supported-Features	TBD1	x.x	Grouped		V
OC-OLR	TBD2	x.x	Grouped		V
OC-Sequence-Number	TBD3	x.x	Unsigned64		V
OC-Validity-Duration	TBD4	x.x	Unsigned32		V
OC-Report-Type	TBD5	x.x	Enumerated		V
OC-Reduction -Percentage	TBD8	x.x	Unsigned32		V
OC-Feature-Vector	TBD6	x.x	Unsigned64		V

As described in the Diameter base protocol [RFC6733], the M-bit setting for a given AVP is relevant to an application and each command within that application that includes the AVP.

The Diameter overload control AVPs SHOULD always be sent with the M-bit cleared when used within existing Diameter applications to avoid backward compatibility issues. Otherwise, when reused in newly defined Diameter applications, the DOIC related AVPs SHOULD have the M-bit set.

#### 7. Error Response Codes

When a DOIC node rejects a Diameter request due to overload, the DOIC node MUST select an appropriate error response code. This



determination is made based on the probability of the request succeeding if retried on a different path.

A reporting node rejecting a Diameter request due to an overload condition SHOULD send a DIAMETER-TOO-BUSY error response, if it can assume that the same request may succeed on a different path.

If a reporting node knows or assumes that the same request will not succeed on a different path, DIAMETER\_UNABLE\_TO\_COMPLY error response SHOULD be used. Retrying would consume valuable resources during an occurrence of overload.

For instance, if the request arrived at the reporting node without a Destination-Host AVP then the reporting node might determine that there is an alternative Diameter node that could successfully process the request and that retrying the transaction would not negatively impact the reporting node. DIAMETER\_TOO\_BUSY would be sent in this case.

For instance, if the request arrived at the reporting node with a Destination-Host AVP populated with its own Diameter identity then the reporting node can assume that retrying the request would result in it coming to the same reporting node. DIAMETER\_UNABLE\_TO\_COMPLY would be sent in this case.

A second example is when an agent that supports the DOIC solution is performing the role of a reacting node for a non supporting client. Requests that are rejected as a result of DOIC throttling by the agent in this scenario would generally be rejected with a DIAMETER\_UNABLE\_TO\_COMPLY response code.

## 8. IANA Considerations

### 8.1. AVP codes

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

### 8.2. New registries

Two new registries are needed under the 'Authentication, Authorization, and Accounting (AAA) Parameters' registry.

Section 6.2 defines a new "Overload Control Feature Vector" registry including the initial assignments. New values can be added into the registry using the Specification Required policy [RFC5226]. See Section 6.2 for the initial assignment in the registry.

Section 6.6 defines a new "Overload Report Type" registry with its initial assignments. New types can be added using the Specification Required policy [RFC5226].

## 9. Security Considerations

This mechanism gives Diameter nodes the ability to request that downstream nodes send fewer Diameter requests. Nodes do this by exchanging overload reports that directly affect this reduction. This exchange is potentially subject to multiple methods of attack, and has the potential to be used as a Denial-of-Service (DoS) attack vector.

Overload reports may contain information about the topology and current status of a Diameter network. This information is potentially sensitive. Network operators may wish to control disclosure of overload reports to unauthorized parties to avoid its use for competitive intelligence or to target attacks.

Diameter does not include features to provide end-to-end authentication, integrity protection, or confidentiality. This may cause complications when sending overload reports between non-adjacent nodes.

### 9.1. Potential Threat Modes

The Diameter protocol involves transactions in the form of requests and answers exchanged between clients and servers. These clients and servers may be peers, that is, they may share a direct transport (e.g. TCP or SCTP) connection, or the messages may traverse one or more intermediaries, known as Diameter Agents. Diameter nodes use TLS, DTLS, or IPSec to authenticate peers, and to provide confidentiality and integrity protection of traffic between peers. Nodes can make authorization decisions based on the peer identities authenticated at the transport layer.

When agents are involved, this presents an effectively hop-by-hop trust model. That is, a Diameter client or server can authorize an agent for certain actions, but it must trust that agent to make appropriate authorization decisions about its peers, and so on.

Since confidentiality and integrity protection occurs at the transport layer. Agents can read, and perhaps modify, any part of a Diameter message, including an overload report.

There are several ways an attacker might attempt to exploit the overload control mechanism. An unauthorized third party might inject an overload report into the network. If this third party is upstream

of an agent, and that agent fails to apply proper authorization policies, downstream nodes may mistakenly trust the report. This attack is at least partially mitigated by the assumption that nodes include overload reports in Diameter answers but not in requests. This requires an attacker to have knowledge of the original request in order to construct a response. Therefore, implementations SHOULD validate that an answer containing an overload report is a properly constructed response to a pending request prior to acting on the overload report.

A similar attack involves an otherwise authorized Diameter node that sends an inappropriate overload report. For example, a server for the realm "example.com" might send an overload report indicating that a competitor's realm "example.net" is overloaded. If other nodes act on the report, they may falsely believe that "example.net" is overloaded, effectively reducing that realm's capacity. Therefore, it's critical that nodes validate that an overload report received from a peer actually falls within that peer's responsibility before acting on the report or forwarding the report to other peers. For example, an overload report from a peer that applies to a realm not handled by that peer is suspect.

An attacker might use the information in an overload report to assist in certain attacks. For example, an attacker could use information about current overload conditions to time a DoS attack for maximum effect, or use subsequent overload reports as a feedback mechanism to learn the results of a previous or ongoing attack.

## 9.2. Denial of Service Attacks

Diameter overload reports can cause a node to cease sending some or all Diameter requests for an extended period. This makes them a tempting vector for DoS attacks. Furthermore, since Diameter is almost always used in support of other protocols, a DoS attack on Diameter is likely to impact those protocols as well. Therefore, Diameter nodes MUST NOT honor or forward overload reports from unauthorized or otherwise untrusted sources.

## 9.3. Non-Compliant Nodes

When a Diameter node sends an overload report, it cannot assume that all nodes will comply. A non-compliant node might continue to send requests with no reduction in load. Requirement 28 [RFC7068] indicates that the overload control solution cannot assume that all Diameter nodes in a network are necessarily trusted, and that malicious nodes not be allowed to take advantage of the overload control mechanism to get more than their fair share of service.

In the absence of an overload control mechanism, Diameter nodes need to implement strategies to protect themselves from floods of requests, and to make sure that a disproportionate load from one source does not prevent other sources from receiving service. For example, a Diameter server might reject a certain percentage of requests from sources that exceed certain limits. Overload control can be thought of as an optimization for such strategies, where downstream nodes never send the excess requests in the first place. However, the presence of an overload control mechanism does not remove the need for these other protection strategies.

#### 9.4. End-to End-Security Issues

The lack of end-to-end security features makes it far more difficult to establish trust in overload reports that originate from non-adjacent nodes. Any agents in the message path may insert or modify overload reports. Nodes must trust that their adjacent peers perform proper checks on overload reports from their peers, and so on, creating a transitive-trust requirement extending for potentially long chains of nodes. Network operators must determine if this transitive trust requirement is acceptable for their deployments. Nodes supporting Diameter overload control **MUST** give operators the ability to select which peers are trusted to deliver overload reports, and whether they are trusted to forward overload reports from non-adjacent nodes.

The lack of end-to-end confidentiality protection means that any Diameter agent in the path of an overload report can view the contents of that report. In addition to the requirement to select which peers are trusted to send overload reports, operators **MUST** be able to select which peers are authorized to receive reports. A node **MUST** not send an overload report to a peer not authorized to receive it. Furthermore, an agent **MUST** remove any overload reports that might have been inserted by other nodes before forwarding a Diameter message to a peer that is not authorized to receive overload reports.

At the time of this writing, the DIME working group is studying requirements for adding end-to-end security [I-D.ietf-dime-e2e-sec-req] features to Diameter. These features, when they become available, might make it easier to establish trust in non-adjacent nodes for overload control purposes. Readers should be reminded, however, that the overload control mechanism encourages Diameter agents to modify AVPs in, or insert additional AVPs into, existing messages that are originated by other nodes. If end-to-end security is enabled, there is a risk that such modification could violate integrity protection. The details of using any future Diameter end-to-end security mechanism with overload control will

require careful consideration, and are beyond the scope of this document.

## 10. Contributors

The following people contributed substantial ideas, feedback, and discussion to this document:

- o Eric McMurry
- o Hannes Tschofenig
- o Ulrich Wiehe
- o Jean-Jacques Trottin
- o Maria Cruz Bartolome
- o Martin Dolly
- o Nirav Salot
- o Susan Shishufeng

## 11. References

### 11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

### 11.2. Informative References

- [Cx] 3GPP, , "ETSI TS 129 229 V11.4.0", August 2013.

- [I-D.ietf-dime-e2e-sec-req]  
Tschofenig, H., Korhonen, J., Zorn, G., and K. Pillay,  
"Diameter AVP Level Security: Scenarios and Requirements",  
draft-ietf-dime-e2e-sec-req-00 (work in progress),  
September 2013.
- [PCC] 3GPP, , "ETSI TS 123 203 V11.12.0", December 2013.
- [RFC4006] Hakala, H., Mattila, L., Koskinen, J-P., Stura, M., and J.  
Loughney, "Diameter Credit-Control Application", RFC 4006,  
August 2005.
- [RFC5729] Korhonen, J., Jones, M., Morand, L., and T. Tsou,  
"Clarifications on the Routing of Diameter Requests Based  
on the Username and the Realm", RFC 5729, December 2009.
- [RFC7068] McMurtry, E. and B. Campbell, "Diameter Overload Control  
Requirements", RFC 7068, November 2013.
- [S13] 3GPP, , "ETSI TS 129 272 V11.9.0", December 2012.

#### Appendix A. Issues left for future specifications

The base solution for the overload control does not cover all possible use cases. A number of solution aspects were intentionally left for future specification and protocol work.

##### A.1. Additional traffic abatement algorithms

This specification describes only means for a simple loss based algorithm. Future algorithms can be added using the designed solution extension mechanism. The new algorithms need to be registered with IANA. See Sections 6.1 and 8 for the required IANA steps.

##### A.2. Agent Overload

This specification focuses on Diameter endpoint (server or client) overload. A separate extension will be required to outline the handling of the case of agent overload.

##### A.3. New Error Diagnostic AVP

The proposal was made to add a new Error Diagnostic AVP to supplement the error responses to be able to indicate that overload was the reason for the rejection of the message.

## Appendix B. Deployment Considerations

### Non supporting agents

Due to the way that realm-routed requests are handled in Diameter networks, with the server selection for the request done by an agent, it is recommended that deployments enable all agents that do server selection to support the DOIC solution prior to enabling the DOIC solution in the Diameter network.

### Topology hiding interactions

There exist proxies that implement what is referred to as Topology Hiding. This can include cases where the agent modifies the Origin-Host in answer messages. The behavior of the DOIC solution is not well understood when this happens. As such, the DOIC solution does not address this scenario.

## Appendix C. Requirements Conformance Analysis

This section contains the result of an analysis of the DOIC solutions conformance to the requirements defined in [RFC7068].

To be completed.

## Appendix D. Considerations for Applications Integrating the DOIC Solution

This section outlines considerations to be taken into account when integrating the DOIC solution into Diameter applications.

### D.1. Application Classification

The following is a classification of Diameter applications and request types. This discussion is meant to document factors that play into decisions made by the Diameter identity responsible for handling overload reports.

Section 8.1 of [RFC6733] defines two state machines that imply two types of applications, session-less and session-based applications. The primary difference between these types of applications is the lifetime of Session-Ids.

For session-based applications, the Session-Id is used to tie multiple requests into a single session.

The Credit-Control application defined in [RFC4006] is an example of a Diameter session-based application.

In session-less applications, the lifetime of the Session-Id is a single Diameter transaction, i.e. the session is implicitly terminated after a single Diameter transaction and a new Session-Id is generated for each Diameter request.

For the purposes of this discussion, session-less applications are further divided into two types of applications:

Stateless applications:

Requests within a stateless application have no relationship to each other. The 3GPP defined S13 application is an example of a stateless application [S13], where only a Diameter command is defined between a client and a server and no state is maintained between two consecutive transactions.

Pseudo-session applications:

Applications that do not rely on the Session-Id AVP for correlation of application messages related to the same session but use other session-related information in the Diameter requests for this purpose. The 3GPP defined Cx application [Cx] is an example of a pseudo-session application.

The handling of overload reports must take the type of application into consideration, as discussed in Appendix D.2.

## D.2. Application Type Overload Implications

This section discusses considerations for mitigating overload reported by a Diameter entity. This discussion focuses on the type of application. Appendix D.3 discusses considerations for handling various request types when the target server is known to be in an overloaded state.

These discussions assume that the strategy for mitigating the reported overload is to reduce the overall workload sent to the overloaded entity. The concept of applying overload treatment to requests targeted for an overloaded Diameter entity is inherent to this discussion. The method used to reduce offered load is not specified here but could include routing requests to another Diameter entity known to be able to handle them, or it could mean rejecting certain requests. For a Diameter agent, rejecting requests will usually mean generating appropriate Diameter error responses. For a Diameter client, rejecting requests will depend upon the application. For example, it could mean giving an indication to the entity requesting the Diameter service that the network is busy and to try again later.



#### Stateless applications:

By definition there is no relationship between individual requests in a stateless application. As a result, when a request is sent or relayed to an overloaded Diameter entity - either a Diameter Server or a Diameter Agent - the sending or relaying entity can choose to apply the overload treatment to any request targeted for the overloaded entity.

#### Pseudo-session applications:

For pseudo-session applications, there is an implied ordering of requests. As a result, decisions about which requests towards an overloaded entity to reject could take the command code of the request into consideration. This generally means that transactions later in the sequence of transactions should be given more favorable treatment than messages earlier in the sequence. This is because more work has already been done by the Diameter network for those transactions that occur later in the sequence. Rejecting them could result in increasing the load on the network as the transactions earlier in the sequence might also need to be repeated.

#### Session-based applications:

Overload handling for session-based applications must take into consideration the work load associated with setting up and maintaining a session. As such, the entity sending requests towards an overloaded Diameter entity for a session-based application might tend to reject new session requests prior to rejecting intra-session requests. In addition, session ending requests might be given a lower probability of being rejected as rejecting session ending requests could result in session status being out of sync between the Diameter clients and servers. Application designers that would decide to reject mid-session requests will need to consider whether the rejection invalidates the session and any resulting session clean-up procedures.

### D.3. Request Transaction Classification

#### Independent Request:

An independent request is not correlated to any other requests and, as such, the lifetime of the session-id is constrained to an individual transaction.

#### Session-Initiating Request:

A session-initiating request is the initial message that establishes a Diameter session. The ACR message defined in [RFC6733] is an example of a session-initiating request.

#### Correlated Session-Initiating Request:

There are cases when multiple session-initiated requests must be correlated and managed by the same Diameter server. It is notably the case in the 3GPP PCC architecture [PCC], where multiple apparently independent Diameter application sessions are actually correlated and must be handled by the same Diameter server.

#### Intra-Session Request:

An intra session request is a request that uses the same Session-Id than the one used in a previous request. An intra session request generally needs to be delivered to the server that handled the session creating request for the session. The STR message defined in [RFC6733] is an example of an intra-session requests.

#### Pseudo-Session Requests:

Pseudo-session requests are independent requests and do not use the same Session-Id but are correlated by other session-related information contained in the request. There exists Diameter applications that define an expected ordering of transactions. This sequencing of independent transactions results in a pseudo session. The AIR, MAR and SAR requests in the 3GPP defined Cx [Cx] application are examples of pseudo-session requests.

### D.4. Request Type Overload Implications

The request classes identified in Appendix D.3 have implications on decisions about which requests should be throttled first. The following list of request treatment regarding throttling is provided as guidelines for application designers when implementing the Diameter overload control mechanism described in this document. The exact behavior regarding throttling is a matter of local policy, unless specifically defined for the application.

#### Independent requests:

Independent requests can generally be given equal treatment when making throttling decisions, unless otherwise indicated by application requirements or local policy.

#### Session-initiating requests:

Session-initiating requests often represent more work than independent or intra-session requests. Moreover, session-initiating requests are typically followed by other session-related requests. Since the main objective of the overload control is to reduce the total number of requests sent to the overloaded entity, throttling decisions might favor allowing intra-session requests over session-initiating requests. In the absence of local policies or application specific requirements to the contrary, Individual session-initiating requests can be given equal treatment when making throttling decisions.

#### Correlated session-initiating requests:

A Request that results in a new binding, where the binding is used for routing of subsequent session-initiating requests to the same server, represents more work load than other requests. As such, these requests might be throttled more frequently than other request types.

#### Pseudo-session requests:

Throttling decisions for pseudo-session requests can take into consideration where individual requests fit into the overall sequence of requests within the pseudo session. Requests that are earlier in the sequence might be throttled more aggressively than requests that occur later in the sequence.

#### Intra-session requests:

There are two types of intra-sessions requests, requests that terminate a session and the remainder of intra-session requests. Implementors and operators may choose to throttle session-terminating requests less aggressively in order to gracefully terminate sessions, allow clean-up of the related resources (e.g. session state) and avoid the need for additional intra-session requests. Favoring session-termination requests may reduce the session management impact on the overloaded entity. The default handling of other intra-session requests might be to treat them equally when making throttling decisions. There might also be application level considerations whether some request types are favored over others.

Authors' Addresses

Jouni Korhonen (editor)  
Broadcom  
Porkkalankatu 24  
Helsinki FIN-00180  
Finland

Email: jouni.nospam@gmail.com

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

Email: srdonovan@usdonovans.com

Ben Campbell  
Oracle  
7460 Warren Parkway  
Frisco, Texas 75034  
United States

Email: ben@nostrum.com

Lionel Morand  
Orange Labs  
38/40 rue du General Leclerc  
Issy-Les-Moulineaux Cedex 9 92794  
France

Phone: +33145296257  
Email: lionel.morand@orange.com

Diameter Maintenance and Extensions (DIME)  
Internet-Draft  
Intended status: Standards Track  
Expires: February 20, 2016

J. Korhonen, Ed.  
Broadcom  
S. Donovan, Ed.  
B. Campbell  
Oracle  
L. Morand  
Orange Labs  
August 19, 2015

Diameter Overload Indication Conveyance  
draft-ietf-dime-ovli-10.txt

Abstract

This specification defines a base solution for Diameter overload control, referred to as Diameter Overload Indication Conveyance (DOIC).

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 February 20, 2016.

Copyright Notice

Copyright (c) 2015 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 . . . . .	5
4. Solution Overview . . . . .	5
4.1. Piggybacking . . . . .	6
4.2. DOIC Capability Announcement . . . . .	7
4.3. DOIC Overload Condition Reporting . . . . .	9
4.4. DOIC Extensibility . . . . .	11
4.5. Simplified Example Architecture . . . . .	11
5. Solution Procedures . . . . .	12
5.1. Capability Announcement . . . . .	12
5.1.1. Reacting Node Behavior . . . . .	13
5.1.2. Reporting Node Behavior . . . . .	13
5.1.3. Agent Behavior . . . . .	14
5.2. Overload Report Processing . . . . .	15
5.2.1. Overload Control State . . . . .	15
5.2.2. Reacting Node Behavior . . . . .	19
5.2.3. Reporting Node Behavior . . . . .	20
5.3. Protocol Extensibility . . . . .	22
6. Loss Algorithm . . . . .	22
6.1. Overview . . . . .	23
6.2. Reporting Node Behavior . . . . .	23
6.3. Reacting Node Behavior . . . . .	24
7. Attribute Value Pairs . . . . .	24
7.1. OC-Supported-Features AVP . . . . .	25
7.2. OC-Feature-Vector AVP . . . . .	25
7.3. OC-OLR AVP . . . . .	25
7.4. OC-Sequence-Number AVP . . . . .	26
7.5. OC-Validity-Duration AVP . . . . .	26
7.6. OC-Report-Type AVP . . . . .	26
7.7. OC-Reduction-Percentage AVP . . . . .	27
7.8. Attribute Value Pair flag rules . . . . .	27
8. Error Response Codes . . . . .	28
9. IANA Considerations . . . . .	28
9.1. AVP codes . . . . .	28
9.2. New registries . . . . .	29
10. Security Considerations . . . . .	29
10.1. Potential Threat Modes . . . . .	30
10.2. Denial of Service Attacks . . . . .	31
10.3. Non-Compliant Nodes . . . . .	31
10.4. End-to End-Security Issues . . . . .	32
11. Contributors . . . . .	33
12. References . . . . .	33

12.1. Normative References . . . . .	33
12.2. Informative References . . . . .	34
Appendix A. Issues left for future specifications . . . . .	34
A.1. Additional traffic abatement algorithms . . . . .	34
A.2. Agent Overload . . . . .	34
A.3. New Error Diagnostic AVP . . . . .	35
Appendix B. Deployment Considerations . . . . .	35
Appendix C. Considerations for Applications Integrating the DOIC Solution . . . . .	35
C.1. Application Classification . . . . .	35
C.2. Application Type Overload Implications . . . . .	36
C.3. Request Transaction Classification . . . . .	38
C.4. Request Type Overload Implications . . . . .	38
Authors' Addresses . . . . .	40

## 1. Introduction

This specification defines a base solution for Diameter overload control, referred to as Diameter Overload Indication Conveyance (DOIC), based on the requirements identified in [RFC7068].

This specification addresses Diameter overload control between Diameter nodes that support the DOIC solution. The solution, which is designed to apply to existing and future Diameter applications, requires no changes to the Diameter base protocol [RFC6733] and is deployable in environments where some Diameter nodes do not implement the Diameter overload control solution defined in this specification.

A new application specification can incorporate the overload control mechanism specified in this document by making it mandatory to implement for the application and referencing this specification normatively. It is the responsibility of the Diameter application designers to define how overload control mechanisms works on that application.

Note that the overload control solution defined in this specification does not address all the requirements listed in [RFC7068]. A number of overload control related features are left for future specifications. See Appendix A for a list of extensions that are currently being considered.

## 2. Terminology and Abbreviations

### Abatement

Reaction to receipt of an overload report resulting in a reduction in traffic sent to the reporting node. Abatement actions include diversion and throttling.

### Abatement Algorithm

An extensible method requested by reporting nodes and used by reacting nodes to reduce the amount of traffic sent during an occurrence of overload control.

### Diversion

An overload abatement treatment where the reacting node selects alternate destinations or paths for requests.

### Host-Routed Requests

Requests that a reacting node knows will be served by a particular host, either due to the presence of a Destination-Host Attribute Value Pair (AVP), or by some other local knowledge on the part of the reacting node.

### Overload Control State (OCS)

Internal state maintained by a reporting or reacting node describing occurrences of overload control.

### Overload Report (OLR)

Overload control information for a particular overload occurrence sent by a reporting node.

### Reacting Node

A Diameter node that acts upon an overload report.

### Realm-Routed Requests

Requests that a reacting node does not know which host will service the request.

### Reporting Node

A Diameter node that generates an overload report. (This may or may not be the overloaded node.)

### Throttling

An abatement treatment that limits the number of requests sent by the reacting node. Throttling can include a Diameter Client choosing to not send requests, or a Diameter Agent or Server rejecting requests with appropriate error responses. In both



cases the result of the throttling is a permanent rejection of the transaction.

### 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. Solution Overview

The Diameter Overload Information Conveyance (DOIC) solution allows Diameter nodes to request other Diameter nodes to perform overload abatement actions, that is, actions to reduce the load offered to the overloaded node or realm.

A Diameter node that supports DOIC is known as a "DOIC node". Any Diameter node can act as a DOIC node, including Diameter Clients, Diameter Servers, and Diameter Agents. DOIC nodes are further divided into "Reporting Nodes" and "Reacting Nodes." A reporting node requests overload abatement by sending Overload Reports (OLR).

A reacting node acts upon OLRs, and performs whatever actions are needed to fulfill the abatement requests included in the OLRs. A Reporting node may report overload on its own behalf, or on behalf of other nodes. Likewise, a reacting node may perform overload abatement on its own behalf, or on behalf of other nodes.

A Diameter node's role as a DOIC node is independent of its Diameter role. For example, Diameter Agents may act as DOIC nodes, even though they are not endpoints in the Diameter sense. Since Diameter enables bi-directional applications, where Diameter Servers can send requests towards Diameter Clients, a given Diameter node can simultaneously act as both a reporting node and a reacting node.

Likewise, a Diameter Agent may act as a reacting node from the perspective of upstream nodes, and a reporting node from the perspective of downstream nodes.

DOIC nodes do not generate new messages to carry DOIC related information. Rather, they "piggyback" DOIC information over existing Diameter messages by inserting new AVPs into existing Diameter requests and responses. Nodes indicate support for DOIC, and any

needed DOIC parameters, by inserting an OC-Supported-Features AVP (Section 7.2) into existing requests and responses. Reporting nodes send OLRs by inserting OC-OLR AVPs (Section 7.3).

A given OLR applies to the Diameter realm and application of the Diameter message that carries it. If a reporting node supports more than one realm and/or application, it reports independently for each combination of realm and application. Similarly, the OC-Supported-Features AVP applies to the realm and application of the enclosing message. This implies that a node may support DOIC for one application and/or realm, but not another, and may indicate different DOIC parameters for each application and realm for which it supports DOIC.

Reacting nodes perform overload abatement according to an agreed-upon abatement algorithm. An abatement algorithm defines the meaning of some of the parameters of an OLR and the procedures required for overload abatement. An overload abatement algorithm separates Diameter requests into two sets. The first set contains the requests that are to undergo overload abatement treatment of either throttling or diversion. The second set contains the requests that are to be given normal routing treatment. This document specifies a single must-support algorithm, namely the "loss" algorithm (Section 6). Future specifications may introduce new algorithms.

Overload conditions may vary in scope. For example, a single Diameter node may be overloaded, in which case reacting nodes may attempt to send requests to other destinations. On the other hand, an entire Diameter realm may be overloaded, in which case such attempts would do harm. DOIC OLRs have a concept of "report type" (Section 7.6), where the type defines such behaviors. Report types are extensible. This document defines report types for overload of a specific host, and for overload of an entire realm.

DOIC works through non supporting Diameter Agents that properly pass unknown AVPs unchanged.

#### 4.1. Piggybacking

There is no new Diameter application defined to carry overload related AVPs. The overload control AVPs defined in this specification have been designed to be piggybacked on top of existing application messages. This is made possible by adding the optional overload control AVPs OC-OLR and OC-Supported-Features into existing commands.

Reacting nodes indicate support for DOIC by including the OC-Supported-Features AVP in all request messages originated or relayed by the reacting node.

Reporting nodes indicate support for DOIC by including the OC-Supported-Features AVP in all answer messages originated or relayed by the reporting node that are in response to a request that contained the OC-Supported-Features AVP. Reporting nodes may include overload reports using the OC-OLR AVP in answer messages.

Note that the overload control solution does not have fixed server and client roles. The DOIC node role is determined based on the message type: whether the message is a request (i.e., sent by a "reacting node") or an answer (i.e., sent by a "reporting node"). Therefore, in a typical "client-server" deployment, the Diameter Client may report its overload condition to the Diameter Server for any Diameter Server initiated message exchange. An example of such is the Diameter Server requesting a re-authentication from a Diameter Client.

#### 4.2. DOIC Capability Announcement

The DOIC solution supports the ability for Diameter nodes to determine if other nodes in the path of a request support the solution. This capability is referred to as DOIC Capability Announcement (DCA) and is separate from Diameter Capability Exchange.

The DCA mechanism uses the OC-Supported-Features AVPs to indicate the Diameter overload features supported.

The first node in the path of a Diameter request that supports the DOIC solution inserts the OC-Supported-Features AVP in the request message.

The individual features supported by the DOIC nodes are indicated in the OC-Feature-Vector AVP. Any semantics associated with the features will be defined in extension specifications that introduce the features.

Note: As discussed elsewhere in the document, agents in the path of the request can modify the OC-Supported-Features AVP.

Note: The DOIC solution must support deployments where Diameter Clients and/or Diameter Servers do not support the DOIC solution. In this scenario, Diameter Agents that support the DOIC solution may handle overload abatement for the non-supporting Diameter nodes. In this case the DOIC agent will insert the OC-Supported-Features AVP in requests that do not already contain one, telling

the reporting node that there is a DOIC node that will handle overload abatement. For transactions where there was an OC-Supporting-Features AVP in the request, the agent will insert the OC-Supported-Features AVP in answers, telling the reacting node that there is a reporting node.

The OC-Feature-Vector AVP will always contain an indication of support for the loss overload abatement algorithm defined in this specification (see Section 6). This ensures that a reporting node always supports at least one of the advertized abatement algorithms received in a request messages.

The reporting node inserts the OC-Supported-Features AVP in all answer messages to requests that contained the OC-Supported-Features AVP. The contents of the reporting node's OC-Supported-Features AVP indicate the set of Diameter overload features supported by the reporting node. This specification defines one exception - the reporting node only includes an indication of support for one overload abatement algorithm, independent of the number of overload abatement algorithms actually supported by the reacting node. The overload abatement algorithm indicated is the algorithm that the reporting node intends to use should it enter an overload condition. Reacting nodes can use the indicated overload abatement algorithm to prepare for possible overload reports and must use the indicated overload abatement algorithm if traffic reduction is actually requested.

Note that the loss algorithm defined in this document is a stateless abatement algorithm. As a result it does not require any actions by reacting nodes prior to the receipt of an overload report. Stateful abatement algorithms that base the abatement logic on a history of request messages sent might require reacting nodes to maintain state in advance of receiving an overload report to ensure that the overload reports can be properly handled.

While it should only be done in exceptional circumstances and not during an active occurrence of overload, a reacting node that wishes to transition to a different abatement algorithm can stop advertising support for the algorithm indicated by the reporting node, as long as support for the loss algorithm is always advertised.

The DCA mechanism must also allow the scenario where the set of features supported by the sender of a request and by agents in the path of a request differ. In this case, the agent can update the OC-Supported-Features AVP to reflect the mixture of the two sets of supported features.

Note: The logic to determine if the content of the OC-Supported-Features AVP should be changed is out-of-scope for this document, as is the logic to determine the content of a modified OC-Supported-Features AVP. These are left to implementation decisions. Care must be taken not to introduce interoperability issues for downstream or upstream DOIC nodes. As such, the agent must act as a fully compliant reporting node to the downstream reacting node and as a fully compliant reacting node to the upstream reporting node.

#### 4.3. DOIC Overload Condition Reporting

As with DOIC capability announcement, overload condition reporting uses new AVPs (Section 7.3) to indicate an overload condition.

The OC-OLR AVP is referred to as an overload report. The OC-OLR AVP includes the type of report, a sequence number, the length of time that the report is valid and abatement algorithm specific AVPs.

Two types of overload reports are defined in this document: host reports and realm reports.

A report of type "HOST\_REPORT" is sent to indicate the overload of a specific host, identified by the Origin-Host AVP of the message containing the OLR, for the application-id indicated in the transaction. When receiving an OLR of type "HOST\_REPORT", a reacting node applies overload abatement treatment to the host-routed requests identified by the overload abatement algorithm (see definition in Section 2) sent for this application to the overloaded host.

A report of type "REALM\_REPORT" is sent to indicate the overload of a realm for the application-id indicated in the transaction. The overloaded realm is identified by the Destination-Realm AVP of the message containing the OLR. When receiving an OLR of type "REALM\_REPORT", a reacting node applies overload abatement treatment to realm-routed requests identified by the overload abatement algorithm (see definition in Section 2) sent for this application to the overloaded realm.

This document assumes that there is a single source for realm-reports for a given realm, or that if multiple nodes can send realm reports, that each such node has full knowledge of the overload state of the entire realm. A reacting node cannot distinguish between receiving realm-reports from a single node, or from multiple nodes.

Note: Known issues exist if multiple sources for overload reports which apply to the same Diameter entity exist. Reacting nodes have no way of determining the source and, as such, will treat

them as coming from a single source. Variance in sequence numbers between the two sources can then cause incorrect overload abatement treatment to be applied for indeterminate periods of time.

Reporting nodes are responsible for determining the need for a reduction of traffic. The method for making this determination is implementation specific and depends on the type of overload report being generated. A host-report might be generated by tracking use of resources required by the host to handle transactions for the Diameter application. A realm-report generally impacts the traffic sent to multiple hosts and, as such, requires tracking the capacity of all servers able to handle realm-routed requests for the application and realm.

Once a reporting node determines the need for a reduction in traffic, it uses the DOIC defined AVPs to report on the condition. These AVPs are included in answer messages sent or relayed by the reporting node. The reporting node indicates the overload abatement algorithm that is to be used to handle the traffic reduction in the OC-Supported-Features AVP. The OC-OLR AVP is used to communicate information about the requested reduction.

Reacting nodes, upon receipt of an overload report, apply the overload abatement algorithm to traffic impacted by the overload report. The method used to determine the requests that are to receive overload abatement treatment is dependent on the abatement algorithm. The loss abatement algorithm is defined in this document (Section 6). Other abatement algorithms can be defined in extensions to the DOIC solution.

Two types of overload abatement treatment are defined, diversion and throttling. Reacting nodes are responsible for determining which treatment is appropriate for individual requests.

As the conditions that lead to the generation of the overload report change the reporting node can send new overload reports requesting greater reduction if the condition gets worse or less reduction if the condition improves. The reporting node sends an overload report with a duration of zero to indicate that the overload condition has ended and abatement is no longer needed.

The reacting node also determines when the overload report expires based on the OC-Validity-Duration AVP in the overload report and stops applying the abatement algorithm when the report expires.

Note that erroneous overload reports can be used for DoS attacks. This includes the ability to indicate that a significant reduction in

traffic, up to and including a request for no traffic, should be sent to a reporting node. As such, care should be taken to verify the sender of overload reports.

#### 4.4. DOIC Extensibility

The DOIC solution is designed to be extensible. This extensibility is based on existing Diameter based extensibility mechanisms, along with the DOIC capability announcement mechanism.

There are multiple categories of extensions that are expected. This includes the definition of new overload abatement algorithms, the definition of new report types and the definition of new scopes of messages impacted by an overload report.

A DOIC node communicates supported features by including them in the OC-Feature-Vector AVP, as a sub-AVP of OC-Supported-Features. Any non-backwards compatible DOIC extensions define new values for the OC-Feature-Vector AVP. DOIC extensions also have the ability to add new AVPs to the OC-Supported-Features AVP, if additional information about the new feature is required.

Overload reports can also be extended by adding new sub-AVPs to the OC-OLR AVP, allowing reporting nodes to communicate additional information about handling an overload condition.

If necessary, new extensions can also define new AVPs that are not part of the OC-Supported-Features and OC-OLR group AVPs. It is, however, recommended that DOIC extensions use the OC-Supported-Features AVP and OC-OLR AVP to carry all DOIC related AVPs.

#### 4.5. Simplified Example Architecture

Figure 1 illustrates the simplified architecture for Diameter overload information conveyance.

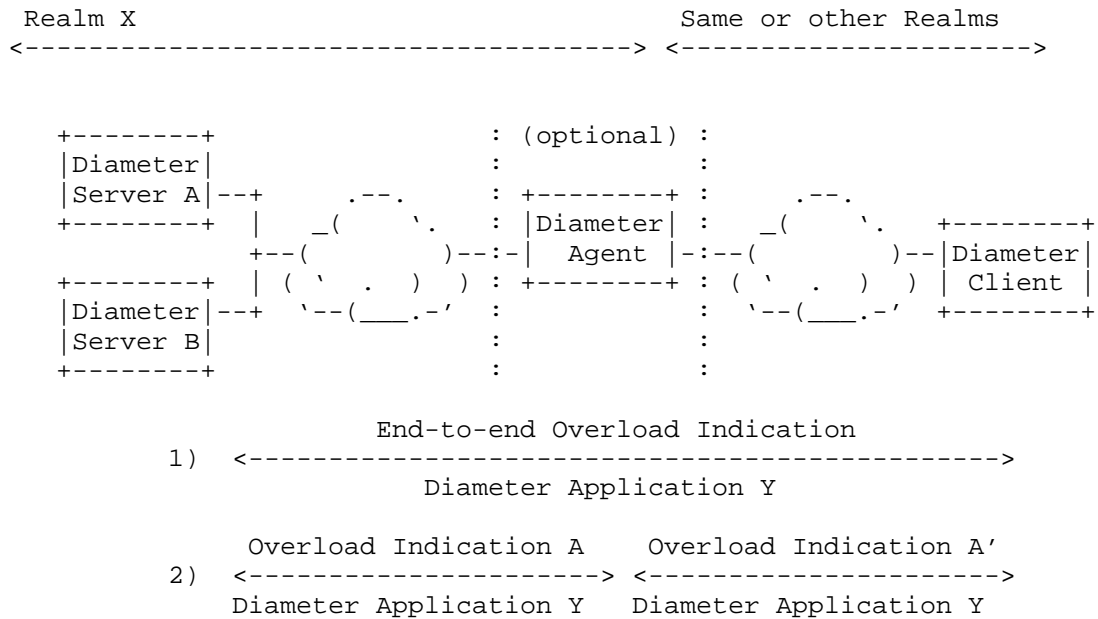


Figure 1: Simplified architecture choices for overload indication delivery

In Figure 1, the Diameter overload indication can be conveyed (1) end-to-end between servers and clients or (2) between servers and Diameter agent inside the realm and then between the Diameter agent and the clients.

## 5. Solution Procedures

This section outlines the normative behavior for the DOIC solution.

### 5.1. Capability Announcement

This section defines DOIC Capability Announcement (DCA) behavior.

Note: This specification assumes that changes in DOIC node capabilities are relatively rare events that occur as a result of administrative action. Reacting nodes ought to minimize changes that force the reporting node to change the features being used, especially during active overload conditions. But even if reacting nodes avoid such changes, reporting nodes still have to be prepared for them to occur. For example, differing capabilities between multiple reacting nodes may still force a



reporting node to select different features on a per-transaction basis.

#### 5.1.1. Reacting Node Behavior

A reacting node MUST include the OC-Supported-Features AVP in all requests. It MAY include the OC-Feature-Vector AVP, as a sub-avp of OC-Supported-Features. If it does so, it MUST indicate support for the "loss" algorithm. If the reacting node is configured to support features (including other algorithms) in addition to the loss algorithm, it MUST indicate such support in an OC-Feature-Vector AVP.

An OC-Supported-Features AVP in answer messages indicates there is a reporting node for the transaction. The reacting node MAY take action, for example creating state for some stateful abatement algorithm, based on the features indicated in the OC-Feature-Vector AVP.

Note: The loss abatement algorithm does not require stateful behavior when there is no active overload report.

Reacting nodes need to be prepared for the reporting node to change selected algorithms. This can happen at any time, including when the reporting node has sent an active overload report. The reacting node can minimize the potential for changes by modifying the advertised abatement algorithms sent to an overloaded reporting node to the currently selected algorithm and loss (or just loss if it is the currently selected algorithm). This has the effect of limiting the potential change in abatement algorithm from the currently selected algorithm to loss, avoiding changes to more complex abatement algorithms that require state to operate properly.

#### 5.1.2. Reporting Node Behavior

Upon receipt of a request message, a reporting node determines if there is a reacting node for the transaction based on the presence of the OC-Supported-Features AVP in the request message.

If the request message contains an OC-Supported-Features AVP then a reporting node MUST include the OC-Supported-Features AVP in the answer message for that transaction.

Note: Capability announcement is done on a per transaction basis. The reporting node cannot assume that the capabilities announced by a reacting node will be the same between transactions.

A reporting node MUST NOT include the OC-Supported-Features AVP, OC-OLR AVP or any other overload control AVPs defined in extension

drafts in response messages for transactions where the request message does not include the OC-Supported-Features AVP. Lack of the OC-Supported-Features AVP in the request message indicates that there is no reacting node for the transaction.

A reporting node knows what overload control functionality is supported by the reacting node based on the content or absence of the OC-Feature-Vector AVP within the OC-Supported-Features AVP in the request message.

A reporting node MUST select a single abatement algorithm in the OC-Feature-Vector AVP. The abatement algorithm selected MUST indicate the abatement algorithm the reporting node wants the reacting node to use when the reporting node enters an overload condition.

The abatement algorithm selected MUST be from the set of abatement algorithms contained in the request message's OC-Feature-Vector AVP.

A reporting node that selects the loss algorithm may do so by including the OC-Feature-Vector AVP with an explicit indication of the loss algorithm, or it MAY omit OC-Feature-Vector. If it selects a different algorithm, it MUST include the OC-Feature-Vector AVP with an explicit indication of the selected algorithm.

The reporting node SHOULD indicate support for other DOIC features defined in extension drafts that it supports and that apply to the transaction. It does so using the OC-Feature-Vector AVP.

Note: Not all DOIC features will apply to all Diameter applications or deployment scenarios. The features included in the OC-Feature-Vector AVP are based on local reporting node policy.

#### 5.1.3. Agent Behavior

Diameter Agents that support DOIC can ensure that all messages relayed by the agent contain the OC-Supported-Features AVP.

A Diameter Agent MAY take on reacting node behavior for Diameter endpoints that do not support the DOIC solution. A Diameter Agent detects that a Diameter endpoint does not support DOIC reacting node behavior when there is no OC-Supported-Features AVP in a request message.

For a Diameter Agent to be a reacting node for a non-supporting Diameter endpoint, the Diameter Agent MUST include the OC-Supported-Features AVP in request messages it relays that do not contain the OC-Supported-Features AVP.

A Diameter Agent MAY take on reporting node behavior for Diameter endpoints that do not support the DOIC solution. The Diameter Agent MUST have visibility to all traffic destined for the non-supporting host in order to become the reporting node for the Diameter endpoint. A Diameter Agent detects that a Diameter endpoint does not support DOIC reporting node behavior when there is no OC-Supported-Features AVP in an answer message for a transaction that contained the OC-Supported-Features AVP in the request message.

If a request already has the OC-Supported-Features AVP, a Diameter agent MAY modify it to reflect the features appropriate for the transaction. Otherwise, the agent relays the OC-Supported-Features AVP without change.

For instance, if the agent supports a superset of the features reported by the reacting node then the agent might choose, based on local policy, to advertise that superset of features to the reporting node.

If the Diameter Agent changes the OC-Supported-Features AVP in a request message then it is likely it will also need to modify the OC-Supported-Features AVP in the answer message for the transaction. A Diameter Agent MAY modify the OC-Supported-Features AVP carried in answer messages.

When making changes to the OC-Supported-Features or OC-OLR AVPs, the Diameter Agent needs to ensure consistency in its behavior with both upstream and downstream DOIC nodes.

## 5.2. Overload Report Processing

### 5.2.1. Overload Control State

Both reacting and reporting nodes maintain Overload Control State (OCS) for active overload conditions. The following sections define behavior associated with that OCS.

The contents of the OCS in the reporting node and in the reacting node represent logical constructs. The actual internal physical structure of the state included in the OCS is an implementation decision.

#### 5.2.1.1. Overload Control State for Reacting Nodes

A reacting node maintains the following OCS per supported Diameter application:

- o A host-type OCS entry for each Destination-Host to which it sends host-type requests and
- o A realm-type OCS entry for each Destination-Realm to which it sends realm-type requests.

A host-type OCS entry is identified by the pair of application-id and the node's DiameterIdentity.

A realm-type OCS entry is identified by the pair of application-id and realm.

The host-type and realm-type OCS entries include the following information (the actual information stored is an implementation decision):

- o Sequence number (as received in OC-OLR, see Section 7.3)
- o 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)
- o Selected Abatement Algorithm (as received in the OC-Supported-Features AVP)
- o Abatement Algorithm specific input data (as received in the OC-OLR AVP, for example, OC-Reduction-Percentage for the Loss abatement algorithm)

#### 5.2.1.2. Overload Control State for Reporting Nodes

A reporting node maintains OCS entries per supported Diameter application, per supported (and eventually selected) Abatement Algorithm and per report-type.

An OCS entry is identified by the tuple of Application-Id, Report-Type and Abatement Algorithm and includes the following information (the actual information stored is an implementation decision):

- o Sequence number
- o Validity Duration
- o Expiration Time
- o Algorithm specific input data (for example, the Reduction Percentage for the Loss Abatement Algorithm)

#### 5.2.1.3. Reacting Node Maintenance of Overload Control State

When a reacting node receives an OC-OLR AVP, it MUST determine if it is for an existing or new overload condition.

Note: For the remainder of this section the term OLR refers to the combination of the contents of the received OC-OLR AVP and the abatement algorithm indicated in the received OC-Supported-Features AVP.

When receiving an answer message with multiple OLRs of different supported report types, a reacting node MUST process each received OLR.

The OLR is for an existing overload condition if a reacting node has an OCS that matches the received OLR.

For a host-report this means it matches the application-id and the host's DiameterIdentity in an existing host OCS entry.

For a realm-report this means it matches the application-id and the realm in an existing realm OCS entry.

If the OLR is for an existing overload condition then a reacting node MUST determine if the OLR is a retransmission or an update to the existing OLR.

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

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

If the reacting node determines that the sequence number has rolled over then the reacting node MUST update the matching OCS entry. This can be determined by recognizing that the number has changed from something close to the maximum value in the OC-Sequence-Number AVP to something close to the minimum value in the OC-Sequence-Number AVP.

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

For a host-report this means a reacting node creates an OCS entry with the application-id in the received message and DiameterIdentity of the Origin-Host in the received message.

Note: This solution assumes that the Origin-Host AVP in the answer message included by the reporting node is not changed along the path to the reacting node.

For a realm-report this means a reacting node creates an OCS entry with the application-id in the received message and realm of the Origin-Realm in the received message.

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

Note: It is not necessarily appropriate to delete the OCS entry, as there is recommended behavior that the reacting node slowly returns to full traffic when ending an overload abatement period.

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").

#### 5.2.1.4. Reporting Node Maintenance of Overload Control State

A reporting node SHOULD create a new OCS entry when entering an overload condition.

Note: If a reporting node knows through absence of the OC-Supported-Features AVP in received messages that there are no reacting nodes supporting DOIC then the reporting node can choose to not create OCS entries.

When generating a new OCS entry the sequence number SHOULD be set to zero ("0").

When generating sequence numbers for new overload conditions, the new sequence number MUST be greater than any sequence number in an active (unexpired) overload report for the same application and report-type previously sent by the reporting node. This property MUST hold over a reboot of the reporting node.

Note: One way of addressing this over a reboot of a reporting node is to use a time stamp for the first overload condition that occurs after the report and to start using sequences beginning with zero for subsequent overload conditions.

A reporting node MUST update an OCS entry when it needs to adjust the validity duration of the overload condition at reacting nodes.

For instance, if a reporting node wishes to instruct reacting nodes to continue overload abatement for a longer period of time

than originally communicated. This also applies if the reporting node wishes to shorten the period of time that overload abatement is to continue.

A reporting node MUST update an OCS entry when it wishes to adjust any abatement algorithm specific parameters, including, for example, the reduction percentage used for the Loss abatement algorithm.

For instance, if a reporting node wishes to change the reduction percentage either higher, if the overload condition has worsened, or lower, if the overload condition has improved, then the reporting node would update the appropriate OCS entry.

A reporting node MUST increment the sequence number associated with the OCS entry anytime the contents of the OCS entry are changed. This will result in a new sequence number being sent to reacting nodes, instructing reacting nodes to process the OC-OLR AVP.

A reporting node SHOULD update an OCS entry with a validity duration of zero ("0") when the overload condition ends.

Note: If a reporting node knows that the OCS entries in the reacting nodes are near expiration then the reporting node might decide not to send an OLR with a validity duration of zero.

A reporting node MUST keep an OCS entry with a validity duration of zero ("0") for a period of time long enough to ensure that any non-expired reacting node's OCS entry created as a result of the overload condition in the reporting node is deleted.

#### 5.2.2. Reacting Node Behavior

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

If the request matches an active OCS then the reacting node MUST use the overload abatement algorithm indicated in the OCS to determine if the request is to receive overload abatement treatment.

For the Loss abatement algorithm defined in this specification, see Section 6 for the overload abatement algorithm logic applied.

If the overload abatement algorithm selects the request for overload abatement treatment then the reacting node MUST apply overload abatement treatment on the request. The abatement treatment applied depends on the context of the request.

If diversion abatement treatment is possible (i.e., a different path for the request can be selected where the overloaded node is not part of the different path), then the reacting node SHOULD apply diversion abatement treatment to the request. The reacting node MUST apply throttling abatement treatment to requests identified for abatement treatment when diversion treatment is not possible or was not applied.

Note: This only addresses the case where there are two defined abatement treatments, diversion and throttling. Any extension that defines a new abatement treatment must also define the interaction of the new abatement treatment with existing treatments.

If the overload abatement treatment results in throttling of the request and if the reacting node is an agent then the agent MUST send an appropriate error as defined in Section 8.

Diameter endpoints that throttle requests need to do so according to the rules of the client application. Those rules will vary by application, and are beyond the scope of this document.

In the case that the OCS entry indicated no traffic was to be sent to the overloaded entity and the validity duration expires then overload abatement associated with the overload report MUST be ended in a controlled fashion.

### 5.2.3. Reporting Node Behavior

If there is an active OCS entry then a reporting node SHOULD include the OC-OLR AVP in all answers to requests that contain the OC-Supported-Features AVP and that match the active OCS entry.

Note: A request matches if the application-id in the request matches the application-id in any active OCS entry and if the report-type in the OCS entry matches a report-type supported by the reporting node as indicated in the OC-Supported-Features AVP.

The contents of the OC-OLR AVP depend on the selected algorithm.

A reporting node MAY choose to not resend an overload report to a reacting node if it can guarantee that this overload report is already active in the reacting node.

Note: In some cases (e.g., when there are one or more agents in the path between reporting and reacting nodes, or when overload reports are discarded by reacting nodes) a reporting node may not



be able to guarantee that the reacting node has received the report.

A reporting node **MUST NOT** send overload reports of a type that has not been advertised as supported by the reacting node.

Note: A reacting node implicitly advertises support for the host and realm report types by including the OC-Supported-Features AVP in the request. Support for other report types will be explicitly indicated by new feature bits in the OC-Feature-Vector AVP.

A reporting node **SHOULD** explicitly indicate the end of an overload occurrence by sending a new OLR with OC-Validity-Duration set to a value of zero ("0"). The reporting node **SHOULD** ensure that all reacting nodes receive the updated overload report.

A reporting node **MAY** rely on the OC-Validity-Duration AVP values for the implicit overload control state cleanup on the reacting node.

Note: All OLRs sent have an expiration time calculated by adding the validity-duration contained in the OLR to the time the message was sent. Transit time for the OLR can be safely ignored. The reporting node can ensure that all reacting nodes have received the OLR by continuing to send it in answer messages until the expiration time for all OLRs sent for that overload condition have expired.

When a reporting node sends an OLR, it effectively delegates any necessary throttling to downstream nodes. If the reporting node also locally throttles the same set of messages, the overall number of throttled requests may be higher than intended. Therefore, before applying local message throttling, a reporting node needs to check if these messages match existing OCS entries, indicating that these messages have survived throttling applied by downstream nodes that have received the related OLR.

However, even if the set of messages match existing OCS entries, the reporting node can still apply other abatement methods such as diversion. The reporting node might also need to throttle requests for reasons other than overload. For example, an agent or server might have a configured rate limit for each client, and throttle requests that exceed that limit, even if such requests had already been candidates for throttling by downstream nodes. The reporting node also has the option to send new OLRs requesting greater reductions in traffic, reducing the need for local throttling.

A reporting node **SHOULD** decrease requested overload abatement treatment in a controlled fashion to avoid oscillations in traffic.

For example, it might wait some period of time after overload ends before terminating the OLR, or it might send a series of OLRs indicating progressively less overload severity.

### 5.3. Protocol Extensibility

The DOIC solution can be extended. Types of potential extensions include new traffic abatement algorithms, new report types or other new functionality.

When defining a new extension that requires new normative behavior, the specification must define a new feature for the OC-Feature-Vector. This feature bit is used to communicate support for the new feature.

The extension may define new AVPs for use in DOIC Capability Announcement and for use in DOIC Overload reporting. These new AVPs SHOULD be defined to be extensions to the OC-Supported-Features or OC-OLR 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.

When defining new report type values, the corresponding specification must define the semantics of the new report types and how they affect the OC-OLR AVP handling.

The OC-Supported-Feature and OC-OLR AVPs can be expanded with optional sub-AVPs only if a legacy DOIC implementation can safely ignore them without breaking backward compatibility for the given OC-Report-Type AVP value. Any new sub-AVPs must not require that the M-bit be set.

Documents that introduce new report types must describe any limitations on their use across non-supporting agents.

As with any Diameter specification, RFC6733 requires all new AVPs to be registered with IANA. See Section 9 for the required procedures. New features (feature bits in the OC-Feature-Vector AVP) and report types (in the OC-Report-Type AVP) MUST be registered with IANA.

## 6. Loss Algorithm

This section documents the Diameter overload loss abatement algorithm.

### 6.1. Overview

The DOIC specification supports the ability for multiple overload abatement algorithms to be specified. The abatement algorithm used for any instance of overload is determined by the Diameter Overload Capability Announcement process documented in Section 5.1.

The loss algorithm described in this section is the default algorithm that must be supported by all Diameter nodes that support DOIC.

The loss algorithm is designed to be a straightforward and stateless overload abatement algorithm. It is used by reporting nodes to request a percentage reduction in the amount of traffic sent. The traffic impacted by the requested reduction depends on the type of overload report.

Reporting nodes request the stateless reduction of the number of requests by an indicated percentage. This percentage reduction is in comparison to the number of messages the node otherwise would send, regardless of how many requests the node might have sent in the past.

From a conceptual level, the logic at the reacting node could be outlined as follows.

1. An overload report is received and the associated OCS is either saved or updated (if required) by the reacting node.
2. A new Diameter request is generated by the application running on the reacting node.
3. The reacting node determines that an active overload report applies to the request, as indicated by the corresponding OCS entry.
4. The reacting node determines if overload abatement treatment should be applied to the request. One approach that could be taken for each request is to select a uniformly selected random number between 1 and 100. If the random number is less than or equal to the indicated reduction percentage then the request is given abatement treatment, otherwise the request is given normal routing treatment.

### 6.2. Reporting Node Behavior

The method a reporting node uses to determine the amount of traffic reduction required to address an overload condition is an implementation decision.

When a reporting node that has selected the loss abatement algorithm determines the need to request a reduction in traffic, it includes an OC-OLR AVP in answer messages as described in Section 5.2.3.

When sending the OC-OLR AVP, the reporting node **MUST** indicate a percentage reduction in the OC-Reduction-Percentage AVP.

The reporting node **MAY** change the reduction percentage in subsequent overload reports. When doing so the reporting node must conform to overload report handling specified in Section 5.2.3.

### 6.3. Reacting Node Behavior

The method a reacting node uses to determine which request messages are given abatement treatment is an implementation decision.

When receiving an OC-OLR in an answer message where the algorithm indicated in the OC-Supported-Features AVP is the loss algorithm, the reacting node **MUST** apply abatement treatment to the requested percentage of request messages sent.

Note: The loss algorithm is a stateless algorithm. As a result, the reacting node does not guarantee that there will be an absolute reduction in traffic sent. Rather, it guarantees that the requested percentage of new requests will be given abatement treatment.

If reacting node comes out of the 100 percent traffic reduction, meaning it has received an OLR indicating that no traffic should be sent, as a result of the overload report timing out the reacting node sending the traffic **SHOULD** be conservative and, for example, first send "probe" messages to learn the overload condition of the overloaded node before converging to any traffic amount/rate decided by the sender. Similar concerns apply in all cases when the overload report times out unless the previous overload report stated 0 percent reduction.

The goal of this behavior is to reduce the probability of overload condition thrashing where an immediate transition from 100% reduction to 0% reduction results in the reporting node moving quickly back into an overload condition.

## 7. Attribute Value Pairs

This section describes the encoding and semantics of the Diameter Overload Indication Attribute Value Pairs (AVPs) defined in this document.

Refer to section 4 of [RFC6733] for more information on AVPs and AVP data types.

### 7.1. OC-Supported-Features AVP

The OC-Supported-Features AVP (AVP code TBD1) is of type Grouped and serves two purposes. First, it announces a node's support for the DOIC solution in general. Second, it contains the description of the supported DOIC features of the sending node. The OC-Supported-Features AVP MUST be included in every Diameter request message a DOIC supporting node sends.

```
OC-Supported-Features ::= < AVP Header: TBD1 >
                        [ OC-Feature-Vector ]
                        * [ AVP ]
```

### 7.2. OC-Feature-Vector AVP

The OC-Feature-Vector AVP (AVP code TBD2) 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.

The OC-Feature-Vector sub-AVP is used to announce the DOIC features supported by the DOIC node, in the form of a flag-bits field in which each bit announces one feature or capability supported by the node. The absence of the OC-Feature-Vector AVP in request messages indicates that only the default traffic abatement algorithm described in this specification is supported. The absence of the OC-Feature-Vector AVP in answer messages indicates that the default traffic abatement algorithm described in this specification is selected (while other traffic abatement algorithms may be supported), and no features other than abatement algorithms are supported.

The following capabilities are defined in this document:

OLR\_DEFAULT\_ALGO (0x0000000000000001)

When this flag is set by the a DOIC reacting node it means that the default traffic abatement (loss) algorithm is supported. When this flag is set by a DOIC reporting node it means that the loss algorithm will be used for requested overload abatement.

### 7.3. OC-OLR AVP

The OC-OLR AVP (AVP code TBD3) is of type Grouped and contains the information necessary to convey an overload report on an overload condition at the reporting node. The application the OC-OLR AVP

applies to is the same as the Application-Id found in the Diameter message header. The host or realm the OC-OLR AVP concerns is determined from the Origin-Host AVP and/or Origin-Realm AVP found in the encapsulating Diameter command. The OC-OLR AVP is intended to be sent only by a reporting node.

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

#### 7.4. OC-Sequence-Number AVP

The OC-Sequence-Number AVP (AVP code TBD4) is of type Unsigned64. Its usage in the context of overload control is described in Section 5.2.

From the functionality point of view, the OC-Sequence-Number AVP is used as a non-volatile increasing counter for a sequence of overload reports between two DOIC nodes for the same overload occurrence. Sequence numbers are treated in a uni-directional manner, i.e., two sequence numbers on each direction between two DOIC nodes are not related or correlated.

#### 7.5. OC-Validity-Duration AVP

The OC-Validity-Duration AVP (AVP code TBD5) is of type Unsigned32 and indicates in seconds the validity time of the overload report. The number of seconds is measured after reception of the first OC-OLR AVP with a given value of OC-Sequence-Number AVP. The default value for the OC-Validity-Duration AVP is 30 seconds. When the OC-Validity-Duration AVP is not present in the OC-OLR AVP, the default value applies. The maximum value for the OC-Validity-Duration AVP is 86,400 seconds (24 hours). If the value received in the OC-Validity-Duration is greater than the maximum value then the default value applies.

#### 7.6. OC-Report-Type AVP

The OC-Report-Type AVP (AVP code TBD6) is of type Enumerated. The value of the AVP describes what the overload report concerns. The following values are initially defined:

HOST\_REPORT 0 The overload report is for a host. Overload abatement treatment applies to host-routed requests.

REALM\_REPORT 1 The overload report is for a realm. Overload abatement treatment applies to realm-routed requests.

#### 7.7. OC-Reduction-Percentage AVP

The OC-Reduction-Percentage AVP (AVP code TBD7) is of type Unsigned32 and describes the percentage of the traffic that the sender is requested to reduce, compared to what it otherwise would send. The OC-Reduction-Percentage AVP applies to the default (loss) algorithm specified in this specification. However, the AVP can be reused for future abatement algorithms, if its semantics fit into the new algorithm.

The value of the Reduction-Percentage AVP is between zero (0) and one hundred (100). Values greater than 100 are ignored. The value of 100 means that all traffic is to be throttled, i.e., the reporting node is under a severe load and ceases to process any new messages. The value of 0 means that the reporting node is in a stable state and has no need for the reacting node to apply any traffic abatement.

#### 7.8. Attribute Value Pair flag rules

Attribute Name	AVP Code	Section Defined	Value Type	AVP flag rules	
				MUST	MUST NOT
OC-Supported-Features	TBD1	7.1	Grouped		V
OC-Feature-Vector	TBD2	7.2	Unsigned64		V
OC-OLR	TBD3	7.3	Grouped		V
OC-Sequence-Number	TBD4	7.4	Unsigned64		V
OC-Validity-Duration	TBD5	7.5	Unsigned32		V
OC-Report-Type	TBD6	7.6	Enumerated		V
OC-Reduction-Percentage	TBD7	7.7	Unsigned32		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. Error Response Codes

When a DOIC node rejects a Diameter request due to overload, the DOIC node MUST select an appropriate error response code. This determination is made based on the probability of the request succeeding if retried on a different path.

Note: This only applies for DOIC nodes that are not the originator of the request.

A reporting node rejecting a Diameter request due to an overload condition SHOULD send a `DIAMETER_TOO_BUSY` error response, if it can assume that the same request may succeed on a different path.

If a reporting node knows or assumes that the same request will not succeed on a different path, `DIAMETER_UNABLE_TO_COMPLY` error response SHOULD be used. Retrying would consume valuable resources during an occurrence of overload.

For instance, if the request arrived at the reporting node without a Destination-Host AVP then the reporting node might determine that there is an alternative Diameter node that could successfully process the request and that retrying the transaction would not negatively impact the reporting node. `DIAMETER_TOO_BUSY` would be sent in this case.

If the request arrived at the reporting node with a Destination-Host AVP populated with its own Diameter identity then the reporting node can assume that retrying the request would result in it coming to the same reporting node. `DIAMETER_UNABLE_TO_COMPLY` would be sent in this case.

A second example is when an agent that supports the DOIC solution is performing the role of a reacting node for a non-supporting client. Requests that are rejected as a result of DOIC throttling by the agent in this scenario would generally be rejected with a `DIAMETER_UNABLE_TO_COMPLY` response code.

## 9. IANA Considerations

### 9.1. AVP codes

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



## 9.2. New registries

Two new registries are needed under the 'Authentication, Authorization, and Accounting (AAA) Parameters' registry.

A new "Overload Control Feature Vector" registry is required. The registry must contain the following:

Feature Vector Value Name

Feature Vector Value

Specification - the specification that defines the new value.

See Section 7.2 for the initial Feature Vector Value in the registry. This specification is the specification defining the value. New values can be added into the registry using the Specification Required policy. [RFC5226].

A new "Overload Report Type" registry is required. The registry must contain the following:

Report Type Value Name

Report Type Value

Specification - the specification that defines the new value.

See Section 7.6 for the initial assignment in the registry. New types can be added using the Specification Required policy [RFC5226].

## 10. Security Considerations

DOIC gives Diameter nodes the ability to request that downstream nodes send fewer Diameter requests. Nodes do this by exchanging overload reports that directly effect this reduction. This exchange is potentially subject to multiple methods of attack, and has the potential to be used as a Denial-of-Service (DoS) attack vector. For instance, a series of injected realm OLRs with a requested reduction percentage of 100% could be used to completely eliminate any traffic from being sent to that realm.

Overload reports may contain information about the topology and current status of a Diameter network. This information is potentially sensitive. Network operators may wish to control disclosure of overload reports to unauthorized parties to avoid its use for competitive intelligence or to target attacks.

Diameter does not include features to provide end-to-end authentication, integrity protection, or confidentiality. This may cause complications when sending overload reports between non-adjacent nodes.

#### 10.1. Potential Threat Modes

The Diameter protocol involves transactions in the form of requests and answers exchanged between clients and servers. These clients and servers may be peers, that is, they may share a direct transport (e.g., TCP or SCTP) connection, or the messages may traverse one or more intermediaries, known as Diameter Agents. Diameter nodes use TLS, DTLS, or IPsec to authenticate peers, and to provide confidentiality and integrity protection of traffic between peers. Nodes can make authorization decisions based on the peer identities authenticated at the transport layer.

When agents are involved, this presents an effectively transitive trust model. That is, a Diameter client or server can authorize an agent for certain actions, but it must trust that agent to make appropriate authorization decisions about its peers, and so on. Since confidentiality and integrity protection occurs at the transport layer, agents can read, and perhaps modify, any part of a Diameter message, including an overload report.

There are several ways an attacker might attempt to exploit the overload control mechanism. An unauthorized third party might inject an overload report into the network. If this third party is upstream of an agent, and that agent fails to apply proper authorization policies, downstream nodes may mistakenly trust the report. This attack is at least partially mitigated by the assumption that nodes include overload reports in Diameter answers but not in requests. This requires an attacker to have knowledge of the original request in order to construct an answer. Such an answer would also need to arrive at a Diameter node via a protected transport connection. Therefore, implementations MUST validate that an answer containing an overload report is a properly constructed response to a pending request prior to acting on the overload report, and that the answer was received via an appropriate transport connection.

A similar attack involves a compromised but otherwise authorized node that sends an inappropriate overload report. For example, a server for the realm "example.com" might send an overload report indicating that a competitor's realm "example.net" is overloaded. If other nodes act on the report, they may falsely believe that "example.net" is overloaded, effectively reducing that realm's capacity. Therefore, it's critical that nodes validate that an overload report received from a peer actually falls within that peer's responsibility

before acting on the report or forwarding the report to other peers. For example, an overload report from a peer that applies to a realm not handled by that peer is suspect. This may require out-of-band, non Diameter agreements and/or mechanisms.

This attack is partially mitigated by the fact that the application, as well as host and realm, for a given OLR is determined implicitly by respective AVPs in the enclosing answer. If a reporting node modifies any of those AVPs, the enclosing transaction will also be affected.

#### 10.2. Denial of Service Attacks

Diameter overload reports, especially realm-reports, can cause a node to cease sending some or all Diameter requests for an extended period. This makes them a tempting vector for DoS attacks. Furthermore, since Diameter is almost always used in support of other protocols, a DoS attack on Diameter is likely to impact those protocols as well. In the worst case, where the Diameter application is being used for access control into an IP network, a coordinated DOS attack could result in the blockage of all traffic into that network. Therefore, Diameter nodes MUST NOT honor or forward OLRs received from peers that are not trusted to send them.

An attacker might use the information in an OLR to assist in DoS attacks. For example, an attacker could use information about current overload conditions to time an attack for maximum effect, or use subsequent overload reports as a feedback mechanism to learn the results of a previous or ongoing attack. Operators need the ability to ensure that OLRs are not leaked to untrusted parties.

#### 10.3. Non-Compliant Nodes

In the absence of an overload control mechanism, Diameter nodes need to implement strategies to protect themselves from floods of requests, and to make sure that a disproportionate load from one source does not prevent other sources from receiving service. For example, a Diameter server might throttle a certain percentage of requests from sources that exceed certain limits. Overload control can be thought of as an optimization for such strategies, where downstream nodes never send the excess requests in the first place. However, the presence of an overload control mechanism does not remove the need for these other protection strategies.

When a Diameter node sends an overload report, it cannot assume that all nodes will comply, even if they indicate support for DOIC. A non-compliant node might continue to send requests with no reduction in load. Such non-compliance could be done accidentally, or

maliciously to gain an unfair advantage over compliant nodes. Requirement 28 [RFC7068] indicates that the overload control solution cannot assume that all Diameter nodes in a network are trusted. It also requires that malicious nodes not be allowed to take advantage of the overload control mechanism to get more than their fair share of service.

#### 10.4. End-to End-Security Issues

The lack of end-to-end integrity features makes it difficult to establish trust in overload reports received from non-adjacent nodes. Any agents in the message path may insert or modify overload reports. Nodes must trust that their adjacent peers perform proper checks on overload reports from their peers, and so on, creating a transitive-trust requirement extending for potentially long chains of nodes. Network operators must determine if this transitive trust requirement is acceptable for their deployments. Nodes supporting Diameter overload control MUST give operators the ability to select which peers are trusted to deliver overload reports, and whether they are trusted to forward overload reports from non-adjacent nodes. DOIC nodes MUST strip DOIC AVPs from messages received from peers that are not trusted for DOIC purposes.

The lack of end-to-end confidentiality protection means that any Diameter agent in the path of an overload report can view the contents of that report. In addition to the requirement to select which peers are trusted to send overload reports, operators MUST be able to select which peers are authorized to receive reports. A node MUST NOT send an overload report to a peer not authorized to receive it. Furthermore, an agent MUST remove any overload reports that might have been inserted by other nodes before forwarding a Diameter message to a peer that is not authorized to receive overload reports.

A DOIC node cannot always automatically detect that a peer also supports DOIC. For example, a node might have a peer that is a non-supporting agent. If nodes on the other side of that agent send OC-Supported-Features AVPs, the agent is likely to forward them as unknown AVPs. Messages received across the non-supporting agent may be indistinguishable from messages received across a DOIC supporting agent, giving the false impression that the non-supporting agent actually supports DOIC. This complicates the transitive-trust nature of DOIC. Operators need to be careful to avoid situations where a non-supporting agent is mistakenly trusted to enforce DOIC related authorization policies.

It is expected that work on end-to-end Diameter security might make it easier to establish trust in non-adjacent nodes for overload control purposes. Readers should be reminded, however, that the

overload control mechanism allows Diameter agents to modify AVPs in, or insert additional AVPs into, existing messages that are originated by other nodes. If end-to-end security is enabled, there is a risk that such modification could violate integrity protection. The details of using any future Diameter end-to-end security mechanism with overload control will require careful consideration, and are beyond the scope of this document.

## 11. Contributors

The following people contributed substantial ideas, feedback, and discussion to this document:

- o Eric McMurry
- o Hannes Tschofenig
- o Ulrich Wiehe
- o Jean-Jacques Trottin
- o Maria Cruz Bartolome
- o Martin Dolly
- o Nirav Salot
- o Susan Shishufeng

## 12. References

### 12.1. 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>>.
- [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>>.

## 12.2. Informative References

- [Cx] 3GPP, , "ETSI TS 129 229 V11.4.0", August 2013.
- [I-D.ietf-dime-e2e-sec-req]  
Tschofenig, H., Korhonen, J., Zorn, G., and K. Pillay,  
"Diameter AVP Level Security: Scenarios and Requirements",  
draft-ietf-dime-e2e-sec-req-01 (work in progress), October  
2013.
- [PCC] 3GPP, , "ETSI TS 123 203 V11.12.0", December 2013.
- [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>>.
- [RFC7068] McMurtry, E. and B. Campbell, "Diameter Overload Control  
Requirements", RFC 7068, DOI 10.17487/RFC7068, November  
2013, <<http://www.rfc-editor.org/info/rfc7068>>.
- [S13] 3GPP, , "ETSI TS 129 272 V11.9.0", December 2012.

## Appendix A. Issues left for future specifications

The base solution for the overload control does not cover all possible use cases. A number of solution aspects were intentionally left for future specification and protocol work. The following subsections define some of the potential extensions to the DOIC solution.

### A.1. Additional traffic abatement algorithms

This specification describes only means for a simple loss based algorithm. Future algorithms can be added using the designed solution extension mechanism. The new algorithms need to be registered with IANA. See Sections 7.1 and 9 for the required IANA steps.

### A.2. Agent Overload

This specification focuses on Diameter endpoint (server or client) overload. A separate extension will be required to outline the handling of the case of agent overload.

### A.3. New Error Diagnostic AVP

This specification indicates the use of existing error messages when nodes reject requests due to overload. There is an expectation that additional error codes or AVPs will be defined in a separate specification to indicate that overload was the reason for the rejection of the message.

## Appendix B. Deployment Considerations

### Non-Supporting Agents

Due to the way that realm-routed requests are handled in Diameter networks with the server selection for the request done by an agent, network operators should enable DOIC at agents that perform server selection first.

### Topology Hiding Interactions

There exist proxies that implement what is referred to as Topology Hiding. This can include cases where the agent modifies the Origin-Host in answer messages. The behavior of the DOIC solution is not well understood when this happens. As such, the DOIC solution does not address this scenario.

### Inter Realm/Administrative Domain Considerations

There are likely to be special considerations for handling DOIC signaling across administrative boundaries. This includes considerations for whether or not information included in the DOIC signaling should be sent across those boundaries. In addition consideration should be taken as to whether or not a reacting node in one realm can be trusted to implement the requested overload abatement handling for overload reports received from a separately administered realm.

## Appendix C. Considerations for Applications Integrating the DOIC Solution

This section outlines considerations to be taken into account when integrating the DOIC solution into Diameter applications.

### C.1. Application Classification

The following is a classification of Diameter applications and request types. This discussion is meant to document factors that play into decisions made by the Diameter entity responsible for handling overload reports.

Section 8.1 of [RFC6733] defines two state machines that imply two types of applications, session-less and session-based applications. The primary difference between these types of applications is the lifetime of Session-Ids.

For session-based applications, the Session-Id is used to tie multiple requests into a single session.

The Credit-Control application defined in [RFC4006] is an example of a Diameter session-based application.

In session-less applications, the lifetime of the Session-Id is a single Diameter transaction, i.e., the session is implicitly terminated after a single Diameter transaction and a new Session-Id is generated for each Diameter request.

For the purposes of this discussion, session-less applications are further divided into two types of applications:

#### Stateless Applications:

Requests within a stateless application have no relationship to each other. The 3GPP defined S13 application is an example of a stateless application [S13], where only a Diameter command is defined between a client and a server and no state is maintained between two consecutive transactions.

#### Pseudo-Session Applications:

Applications that do not rely on the Session-Id AVP for correlation of application messages related to the same session but use other session-related information in the Diameter requests for this purpose. The 3GPP defined Cx application [Cx] is an example of a pseudo-session application.

The handling of overload reports must take the type of application into consideration, as discussed in Appendix C.2.

### C.2. Application Type Overload Implications

This section discusses considerations for mitigating overload reported by a Diameter entity. This discussion focuses on the type of application. Appendix C.3 discusses considerations for handling various request types when the target server is known to be in an overloaded state.

These discussions assume that the strategy for mitigating the reported overload is to reduce the overall workload sent to the



overloaded entity. The concept of applying overload treatment to requests targeted for an overloaded Diameter entity is inherent to this discussion. The method used to reduce offered load is not specified here but could include routing requests to another Diameter entity known to be able to handle them, or it could mean rejecting certain requests. For a Diameter agent, rejecting requests will usually mean generating appropriate Diameter error responses. For a Diameter client, rejecting requests will depend upon the application. For example, it could mean giving an indication to the entity requesting the Diameter service that the network is busy and to try again later.

#### Stateless Applications:

By definition there is no relationship between individual requests in a stateless application. As a result, when a request is sent or relayed to an overloaded Diameter entity - either a Diameter Server or a Diameter Agent - the sending or relaying entity can choose to apply the overload treatment to any request targeted for the overloaded entity.

#### Pseudo-Session Applications:

For pseudo-session applications, there is an implied ordering of requests. As a result, decisions about which requests towards an overloaded entity to reject could take the command code of the request into consideration. This generally means that transactions later in the sequence of transactions should be given more favorable treatment than messages earlier in the sequence. This is because more work has already been done by the Diameter network for those transactions that occur later in the sequence. Rejecting them could result in increasing the load on the network as the transactions earlier in the sequence might also need to be repeated.

#### Session-Based Applications:

Overload handling for session-based applications must take into consideration the work load associated with setting up and maintaining a session. As such, the entity sending requests towards an overloaded Diameter entity for a session-based application might tend to reject new session requests prior to rejecting intra-session requests. In addition, session ending requests might be given a lower probability of being rejected as rejecting session ending requests could result in session status being out of sync between the Diameter clients and servers. Application designers that would decide to reject mid-session

requests will need to consider whether the rejection invalidates the session and any resulting session cleanup procedures.

### C.3. Request Transaction Classification

#### Independent Request:

An independent request is not correlated to any other requests and, as such, the lifetime of the session-id is constrained to an individual transaction.

#### Session-Initiating Request:

A session-initiating request is the initial message that establishes a Diameter session. The ACR message defined in [RFC6733] is an example of a session-initiating request.

#### Correlated Session-Initiating Request:

There are cases when multiple session-initiated requests must be correlated and managed by the same Diameter server. It is notably the case in the 3GPP PCC architecture [PCC], where multiple apparently independent Diameter application sessions are actually correlated and must be handled by the same Diameter server.

#### Intra-Session Request:

An intra-session request is a request that uses the same Session-Id than the one used in a previous request. An intra-session request generally needs to be delivered to the server that handled the session creating request for the session. The STR message defined in [RFC6733] is an example of an intra-session request.

#### Pseudo-Session Requests:

Pseudo-session requests are independent requests and do not use the same Session-Id but are correlated by other session-related information contained in the request. There exists Diameter applications that define an expected ordering of transactions. This sequencing of independent transactions results in a pseudo session. The AIR, MAR and SAR requests in the 3GPP defined Cx [Cx] application are examples of pseudo-session requests.

### C.4. Request Type Overload Implications

The request classes identified in Appendix C.3 have implications on decisions about which requests should be throttled first. The following list of request treatment regarding throttling is provided

as guidelines for application designers when implementing the Diameter overload control mechanism described in this document. The exact behavior regarding throttling is a matter of local policy, unless specifically defined for the application.

#### Independent Requests:

Independent requests can generally be given equal treatment when making throttling decisions, unless otherwise indicated by application requirements or local policy.

#### Session-Initiating Requests:

Session-initiating requests often represent more work than independent or intra-session requests. Moreover, session-initiating requests are typically followed by other session-related requests. Since the main objective of the overload control is to reduce the total number of requests sent to the overloaded entity, throttling decisions might favor allowing intra-session requests over session-initiating requests. In the absence of local policies or application specific requirements to the contrary, Individual session-initiating requests can be given equal treatment when making throttling decisions.

#### Correlated Session-Initiating Requests:

A Request that results in a new binding, where the binding is used for routing of subsequent session-initiating requests to the same server, represents more work load than other requests. As such, these requests might be throttled more frequently than other request types.

#### Pseudo-Session Requests:

Throttling decisions for pseudo-session requests can take into consideration where individual requests fit into the overall sequence of requests within the pseudo session. Requests that are earlier in the sequence might be throttled more aggressively than requests that occur later in the sequence.

#### Intra-Session Requests:

There are two types of intra-sessions requests, requests that terminate a session and the remainder of intra-session requests. Implementers and operators may choose to throttle session-terminating requests less aggressively in order to gracefully terminate sessions, allow cleanup of the related resources (e.g., session state) and avoid the need for additional intra-session

requests. Favoring session-termination requests may reduce the session management impact on the overloaded entity. The default handling of other intra-session requests might be to treat them equally when making throttling decisions. There might also be application level considerations whether some request types are favored over others.

#### Authors' Addresses

Jouni Korhonen (editor)  
Broadcom  
Porkkalankatu 24  
Helsinki FIN-00180  
Finland

Email: jouni.nospam@gmail.com

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

Email: srdonovan@usdonovans.com

Ben Campbell  
Oracle  
7460 Warren Parkway  
Frisco, Texas 75034  
United States

Email: ben@nostrum.com

Lionel Morand  
Orange Labs  
38/40 rue du General Leclerc  
Issy-Les-Moulineaux Cedex 9 92794  
France

Phone: +33145296257  
Email: lionel.morand@orange.com

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: February 3, 2013

G. Zorn  
Network Zen  
Q. Wu  
Huawei  
J. Korhonen  
NSN  
August 2, 2012

Diameter Support for Proxy Mobile IPv6 Localized Routing  
draft-ietf-dime-pmip6-lr-18

Abstract

In Proxy Mobile IPv6, packets received from a Mobile Node (MN) by the Mobile Access Gateway (MAG) to which it is attached are typically tunneled to a Local Mobility Anchor (LMA) for routing. The term "localized routing" refers to a method by which packets are routed directly between an MN's MAG and the MAG of its Correspondent Node (CN) without involving any LMA. In a Proxy Mobile IPv6 deployment, it may be desirable to control the establishment of localized routing sessions between two MAGs in a Proxy Mobile IPv6 domain by requiring that the session be authorized. This document specifies how to accomplish this using the Diameter protocol.

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 February 3, 2013.

Copyright Notice

Copyright (c) 2012 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. Solution Overview . . . . .	3
4. Attribute Value Pair Used in this Document . . . . .	4
4.1. User-Name AVP . . . . .	5
4.2. PMIP6-IPv4-Home-Address AVP . . . . .	5
4.3. MIP6-Home-Link-Prefix AVP . . . . .	5
4.4. MIP6-Feature-Vector AVP . . . . .	5
5. Example Signaling Flows for Localized Routing Service	
Authorization . . . . .	6
6. Security Considerations . . . . .	9
7. IANA Considerations . . . . .	10
8. Contributors . . . . .	10
9. Acknowledgements . . . . .	10
10. References . . . . .	10
10.1. Normative References . . . . .	10
10.2. Informative References . . . . .	11
Authors' Addresses . . . . .	11

## 1. Introduction

Proxy Mobile IPv6 (PMIPv6) [RFC5213] allows the Mobility Access Gateway (MAG) to optimize media delivery by locally routing packets from a Mobile Node to a Correspondent Node that is locally attached to an access link connected to the same Mobile Access Gateway, avoiding tunneling them to the Mobile Node's Local Mobility Anchor (LMA). This is referred to as "local routing" in RFC 5213. However, this mechanism is not applicable to the typical scenarios in which the MN and CN are connected to different MAGs and are registered to the same LMA or different LMAs. [RFC6279] takes those typical scenarios into account and defines the problem statement for PMIPv6 localized routing. [I-D.ietf-netext-pmip-lr] specifies the PMIPv6 localized routing protocol based on the scenarios A11, A12, and A21 [RFC6279], which is used to establish a localized routing path between two Mobile Access Gateways in a PMIPv6 domain.

However, there is no relevant work discussing how AAA-based mechanisms can be used to provide authorization to the Mobile Node's MAG or LMA for enabling localized routing between MAGs.

This document describes Diameter [I-D.ietf-dime-rfc3588bis] support for the authorization of PMIPv6 mobility entities in case of A11, A12, A21 during localized routing.

## 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].

## 3. Solution Overview

This document addresses how to provide authorization to the Mobile Node's MAG or LMA for enabling localized routing and resolve the destination MN's MAG by means of interaction between the LMA and the AAA server. Figure 1 shows the reference architecture for Localized Routing Service Authorization. This reference architecture assumes that

- o If MN and CN belong to different LMAs, MN and CN should share the same MAG (i.e., A12 described in [RFC6279]), e.g., MN1 and CN2 in Figure 1 are attached to the same MAG1 and belong to LMA1 and LMA2 respectively. Note that LMA1 and LMA2 in Figure 1 are in the same provider domain (as described in [RFC6279]).

- o If MN and CN are attached to the different MAGs, MN and CN should belong to the same LMA (i.e., A21 described in [RFC6279]), e.g., MN1 and CN3 in the Figure 1 are attached to the MAG1 and MAG3 respectively but belong to LMA1.
- o MN and CN may belong to the same LMA and are attached to the same MAG (i.e., A11 described in [RFC6279]), e.g., MN1 and CN1 in the Figure 1 are both attached to the MAG1 and belong to LMA1.
- o The MAG and LMA support Diameter client functionality.

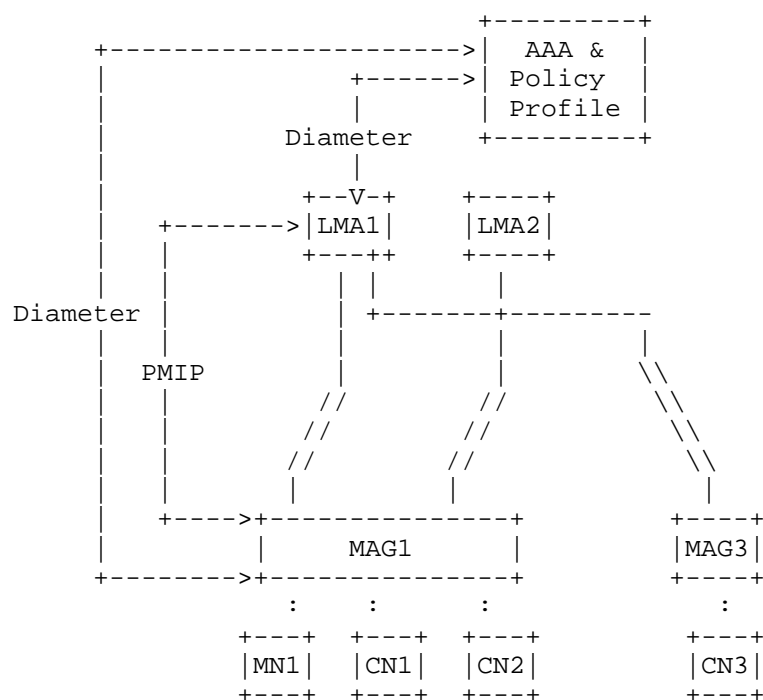


Figure 1: Localized Routing Service Authorization Reference Architecture

The interaction of the MAG and LMA with the AAA server according to the extension specified in this document is used to authorize the localized routing service.

#### 4. Attribute Value Pair Used in this Document

This section describes Attribute Value Pairs (AVPs) defined by this



specification or re-used from existing specifications in a PMIPv6-specific way.

#### 4.1. User-Name AVP

The User-Name AVP (AVP Code 1) is defined in [I-D.ietf-dime-rfc3588bis]. This AVP is used to carry the MN-Identifier (Mobile Node identifier) [RFC5213] in the AA-Request (AAR) message [I-D.ietf-dime-rfc4005bis].

#### 4.2. PMIP6-IPv4-Home-Address AVP

The PMIP6-IPv4-Home-Address AVP (AVP Code 505) is defined in [RFC5779]. This AVP is used to carry the IPv4-MN-HoA (Mobile Node's IPv4 home address) [RFC5844] in the AA-Request (AAR) message [I-D.ietf-dime-rfc4005bis].

#### 4.3. MIP6-Home-Link-Prefix AVP

The MIP6-Home-Link-Prefix AVP (AVP Code 125) is defined in [RFC5779]. This AVP is used to carry the MN-HNP (Mobile Node's home network prefix) in the AAR.

#### 4.4. MIP6-Feature-Vector AVP

The MIP6-Feature-Vector AVP is defined in [RFC5447]. This document allocates a new capability flag bit according to the IANA rules in RFC 5447.

#### INTER\_MAG\_ROUTING\_SUPPORTED (TBD)

Direct routing of IP packets between MNs anchored to different MAGs without involving any LMA is supported. This bit is used with MN-Identifier. When a MAG or LMA sets this bit in the MIP6-Feature-Vector and MN-Identifier corresponding to the Mobile Node is carried with this bit, it indicates to the home AAA server (HAAA) that the Mobile Node associated with this LMA is allowed to use localized routing. If this bit is cleared and MN-Identifier corresponding to the Mobile Node is carried with this bit, it indicates to the home AAA server (HAAA) that the Mobile Node associated with this LMA is not allowed to use localized routing. When a MAG or LMA sets this bit in the MIP6-Feature-Vector and MN-Identifiers corresponding to the Mobile Node and Correspondent Node are both carried with this bit, it indicates to the HAAA that localized routing of IP packets between Mobile Node and Correspondent Node anchored to different MAGs is supported. If this bit is cleared and MN-Identifiers corresponding to the Mobile Node and Correspondent Node are both carried with this bit

to HAAA, it indicates to the HAAA that localized routing of IP packets between Mobile Node and Correspondent Node anchored to different MAGs is not supported. If this bit is cleared in the returned MIP6-Feature-Vector AVP, the HAAA does not authorize direct routing of packets between MNs anchored to the different MAG. The MAG and LMA MUST support this policy feature on a per-MN and per-subscription basis.

## 5. Example Signaling Flows for Localized Routing Service Authorization

Localized Routing Service Authorization can happen during the network access authentication procedure [RFC5779] before localized routing is initialized. In this case, the preauthorized pairs of LMA/prefix sets can be downloaded to Proxy Mobile IPv6 entities during the RFC 5779 procedure. Localized routing can be initiated once the destination of a received packet matches one or more of the prefixes received during the RFC 5779 procedure.

Figure 2 shows an example scenario in which MAG1 acts as a Diameter client, processing the data packet from MN1 to MN2 and requesting authorization of localized routing (i.e., MAG-Initiated LR authorization). In this example scenario, MN1 and MN2 are attached to the same MAG and anchored to the different LMAs (i.e., A12 described in [RFC6279]). In this case, MAG1 knows that MN2 belongs to a different LMA (which can be determined by looking up the binding cache entries corresponding to MN1 and MN2 and comparing the addresses of LMA1 and LMA2). In order to setup a localized routing path with MAG2, MAG1 acts as Diameter client and sends an AAR message to the Diameter server. The message contains an instance of the MIP6-Feature-Vector (MFV) AVP ([RFC5447], Section 4.2.5) with the LOCAL\_MAG\_ROUTING\_SUPPORTED bit ([RFC5779], Section 5.5) set, two instances of the User-Name AVP ([I-D.ietf-dime-rfc3588bis], Section 8.14) containing MN1-Identifier and MN2-Identifier. In addition, the message may contain either an instance of the MIP6-Home-Link-Prefix AVP ([RFC5779], Section 5.3) or an instance of the PMIP6-IPv4-Home-Address AVP ([RFC5779], Section 5.2) containing the IP address/ HNP of MN1.

The Diameter server authorizes localized routing service by checking if MN1 and MN2 are allowed to use localized routing. If so, the Diameter server responds with an AAA message encapsulating an instance of the MIP6-Feature-Vector (MFV) AVP ([RFC5447], Section 4.2.5) with the LOCAL\_MAG\_ROUTING\_SUPPORTED bit ([RFC5779], Section 5.5) set indicating direct routing of IP packets between MNs anchored to the same MAG is supported. MAG1 then knows the localized routing between MN1 and MN2 is allowed. Then MAG1 sends the Request messages respectively to LMA1 and LMA2. The

request message is the Localized Routing Initialization (LRI) message in Figure 2 and belongs to the Initial phase of the localized routing. LMA1 and LMA2 responds to MAG1 using the Localized Routing Acknowledge message (LRA in Figure 2 ) in accordance with [I-D.ietf-netext-pmip-lr].

In case of LRA\_WAIT\_TIME expiration [I-D.ietf-netext-pmip-lr],MAG1 should ask for authorization of localized routing again according to the procedure described above before LRI is retransmitted up to a maximum of LRI\_RETRIES.

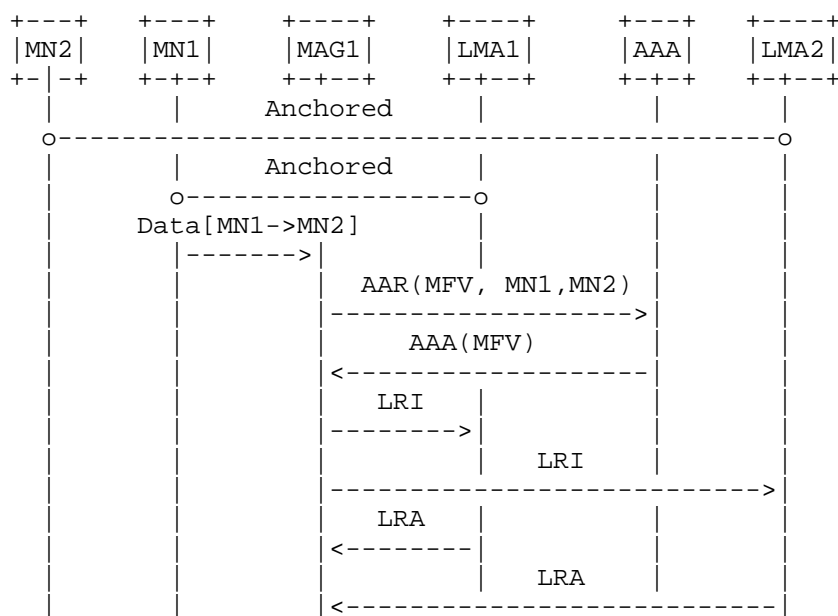


Figure 2: MAG-initiated Localized Routing Authorization in A12

Figure 3 shows the second example scenario, in which LMA1 acts as a Diameter client, processing the data packet from MN2 to MN1 and requesting the authorization of localized routing. In this scenario, MN1 and MN2 are attached to the different MAG and anchored to the same LMA (i.e., A21 described in [RFC6279] ), LMA knows that MN1 and MN2 belong to the same LMA (which can be determined by looking up the binding cache entries corresponding to MN1 and MN2 and comparing the addresses of LMA corresponding to MN1 and LMA corresponding to MN2). In contrast with the signaling flow shown in Figure 2, it is LMA1 instead of MAG1 which initiates the setup of the localized routing path.

The Diameter client in LMA1 sends an AA-Request message to the Diameter server. The message contains an instance of the MIP6-Feature-Vector (MFV) AVP ([RFC5447], Section 4.2.5) with the INTER\_MAG\_ROUTING\_SUPPORTED bit (Section 4.5) set indicating direct routing of IP packets between MNs anchored to different MAGs is supported and two instances of the User-Name AVP ([I-D.ietf-dime-rfc3588bis], Section 8.14) containing MN1-Identifier and MN2-Identifier. The Diameter server authorizes the localized routing service by checking if MN1 and MN2 are allowed to use localized routing. If so, the Diameter server responds with an AA-Answer message encapsulating an instance of the MIP6-Feature-Vector (MFV) AVP ([RFC5447], Section 4.2.5) with the INTER\_MAG\_ROUTING\_SUPPORTED bit (Section 4.5) set indicating direct routing of IP packets between MNs anchored to different MAGs is supported. LMA1 then knows the localized routing is allowed. In success case, LMA1 responds to MAG1 in accordance with [I-D.ietf-netext-pmip-lr].

In case of LRA\_WAIT\_TIME expiration [I-D.ietf-netext-pmip-lr], LMA1 should ask for authorization of localized routing again according to the procedure described above before LRI is retransmitted up to a maximum of LRI\_RETRIES.

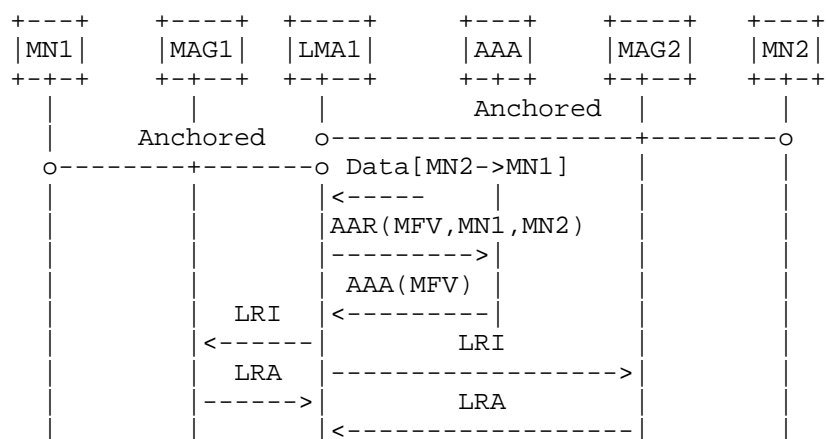


Figure 3: LMA-initiated Localized Routing Authorization in A21

Figure 4 shows another example scenario, in which LMA1 acts as a Diameter client, processing the data packet from MN2 to MN1 and requesting the authorization of localized routing. In this scenario, MN1 and MN2 are attached to the same MAG and anchored to the same LMA (i.e., A11 described in [RFC6279]), LMA knows that MN1 and MN2 belong to the same LMA (which can be determined by looking up the binding

cache entries corresponding to MN1 and MN2 and comparing the addresses of LMA corresponding to MN1 and LMA corresponding to MN2).

The Diameter client in LMA1 sends an AA-Request message to the Diameter server. The message contains an instance of the MIP6-Feature-Vector AVP ([RFC5447], Section 4.2.5) with the LOCAL\_MAG\_ROUTING\_SUPPORTED bit set and two instances of the User-Name AVP ([I-D.ietf-dime-rfc3588bis], Section 8.14) containing MN1-Identifier and MN2-Identifier. The Diameter server authorizes the localized routing service by checking if MN1 and MN2 are allowed to use localized routing. If so, the Diameter server responds with an AA-Answer message encapsulating an instance of the MIP6-Feature-Vector (MFV) AVP ([RFC5447], Section 4.2.5) with the LOCAL\_MAG\_ROUTING\_SUPPORTED bit ([RFC5779], Section 5.5) set indicating direct routing of IP packets between MNs anchored to the same MAG is supported. LMA1 then knows the localized routing is allowed and responds to MAG1 for localized routing in accordance with [I-D.ietf-netext-pmip-lr].

In case of LRA\_WAIT\_TIME expiration [I-D.ietf-netext-pmip-lr], LMA1 should ask for authorization of localized routing again according to the procedure described above before LRI is retransmitted up to a maximum of LRI\_RETRIES.

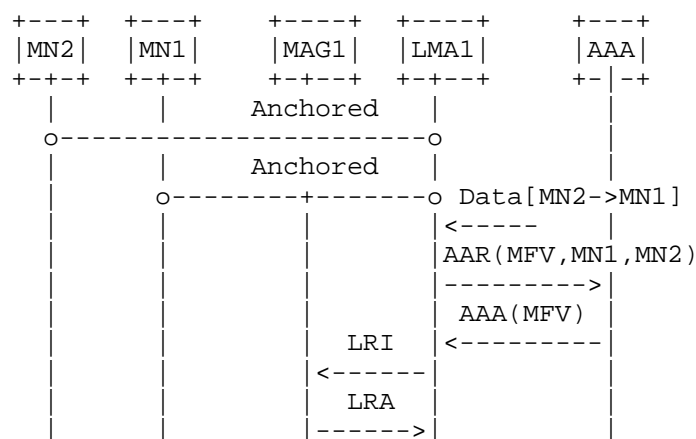


Figure 4: LMA-initiated Localized Routing Authorization in All

## 6. Security Considerations

The security considerations for the Diameter NASREQ [I-D.ietf-dime-rfc4005bis] and Diameter Proxy Mobile IPv6 [RFC5779] applications are also applicable to this document.

The service authorization solicited by the MAG or the LMA relies upon the existing trust relationship between the MAG/LMA and the AAA server.

An authorised MAG could in principle track the movement of any participating CNs at the level of the MAG to which they are anchored. If such a MAG were compromised, or under the control of a bad-actor, then such tracking could represent a privacy breach for the set of tracked CNs. In such a case, the traffic pattern from the compromised MAG might be notable so monitoring for e.g. excessive queries from MAGs might be worthwhile.

## 7. IANA Considerations

This specification defines a new value in the Mobility Capability registry [RFC5447] for use with the MIP6-Feature-Vector AVP: INTER\_MAG\_ROUTING\_SUPPORTED (see Section 4.4).

## 8. Contributors

Paulo Loureiro, Jinwei Xia and Yungui Wang all contributed to early versions of this document.

## 9. Acknowledgements

The authors would like to thank Marco Liebsch, Carlos Jesus Bernardos Cano, Dan Romascanu, Elwyn Davies, Basavaraj Patil, Ralph Droms, Stephen Farrel, Robert Sparks, Benoit Claise and Abhay Roy for their valuable comments and suggestions on this document.

## 10. References

### 10.1. Normative References

- [I-D.ietf-dime-rfc3588bis]  
Fajardo, V., Arkko, J., Loughney, J., and G. Zorn,  
"Diameter Base Protocol", draft-ietf-dime-rfc3588bis-34  
(work in progress), June 2012.
- [I-D.ietf-dime-rfc4005bis]  
Zorn, G., "Diameter Network Access Server Application",  
draft-ietf-dime-rfc4005bis-11 (work in progress),  
July 2012.

- [I-D.ietf-netext-pmip-lr]  
Krishnan, S., Koodli, R., Loureiro, P., Wu, Q., and A. Dutta, "Localized Routing for Proxy Mobile IPv6", draft-ietf-netext-pmip-lr-10 (work in progress), May 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5213] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, August 2008.
- [RFC5447] Korhonen, J., Bournelle, J., Tschofenig, H., Perkins, C., and K. Chowdhury, "Diameter Mobile IPv6: Support for Network Access Server to Diameter Server Interaction", RFC 5447, February 2009.
- [RFC5779] Korhonen, J., Bournelle, J., Chowdhury, K., Muhanna, A., and U. Meyer, "Diameter Proxy Mobile IPv6: Mobile Access Gateway and Local Mobility Anchor Interaction with Diameter Server", RFC 5779, February 2010.
- [RFC5844] Wakikawa, R. and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6", RFC 5844, May 2010.

## 10.2. Informative References

- [RFC6279] Liebsch, M., Jeong, S., and Q. Wu, "Proxy Mobile IPv6 (PMIPv6) Localized Routing Problem Statement", RFC 6279, June 2011.

## Authors' Addresses

Glen Zorn  
Network Zen  
227/358 Thanon Sanphawut  
Bang Na, Bangkok 10260  
Thailand

Phone: +66 (0) 87-040-4617  
Email: glenzorn@gmail.com

Qin Wu  
Huawei Technologies Co., Ltd.  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 21001  
China

Phone: +86-25-84565892  
Email: sunseawq@huawei.com

Jouni Korhonen  
Nokia Siemens Networks  
Linnoitustie 6  
Espoo FI-02600,  
Finland

Email: jouni.nospam@gmail.com





Network Working Group  
Internet-Draft  
Obsoletes: 4005 (if approved)  
Intended status: Standards Track  
Expires: June 1, 2014

G. Zorn, Ed.  
Network Zen  
November 28, 2013

Diameter Network Access Server Application  
draft-ietf-dime-rfc4005bis-14

Abstract

This document describes the Diameter protocol application used for Authentication, Authorization, and Accounting (AAA) services in the Network Access Server (NAS) environment; it obsoletes RFC 4005. When combined with the Diameter Base protocol, Transport Profile, and Extensible Authentication Protocol specifications, this application specification satisfies typical network access services requirements.

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 June 1, 2014.

Copyright Notice

Copyright (c) 2013 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 . . . . .	4
1.1.	Changes from RFC 4005 . . . . .	5
1.2.	Terminology . . . . .	6
1.3.	Requirements Language . . . . .	7
1.4.	Advertising Application Support . . . . .	8
1.5.	Application Identification . . . . .	8
1.6.	Accounting Model . . . . .	8
2.	NAS Calls, Ports, and Sessions . . . . .	8
2.1.	Diameter Session Establishment . . . . .	8
2.2.	Diameter Session Reauthentication or Reauthorization . . . . .	9
2.3.	Diameter Session Termination . . . . .	10
3.	Diameter NAS Application Messages . . . . .	10
3.1.	AA-Request (AAR) Command . . . . .	11
3.2.	AA-Answer (AAA) Command . . . . .	12
3.3.	Re-Auth-Request (RAR) Command . . . . .	14
3.4.	Re-Auth-Answer (RAA) Command . . . . .	15
3.5.	Session-Termination-Request (STR) Command . . . . .	16
3.6.	Session-Termination-Answer (STA) Command . . . . .	17
3.7.	Abort-Session-Request (ASR) Command . . . . .	17
3.8.	Abort-Session-Answer (ASA) Command . . . . .	18
3.9.	Accounting-Request (ACR) Command . . . . .	19
3.10.	Accounting-Answer (ACA) Command . . . . .	21
4.	Diameter NAS Application AVPs . . . . .	22
4.1.	Derived AVP Data Formats . . . . .	22
4.1.1.	QoSFilterRule . . . . .	22
4.2.	NAS Session AVPs . . . . .	23
4.2.1.	Call and Session Information . . . . .	24
4.2.2.	NAS-Port AVP . . . . .	24
4.2.3.	NAS-Port-Id AVP . . . . .	25
4.2.4.	NAS-Port-Type AVP . . . . .	25
4.2.5.	Called-Station-Id AVP . . . . .	25
4.2.6.	Calling-Station-Id AVP . . . . .	25
4.2.7.	Connect-Info AVP . . . . .	26
4.2.8.	Originating-Line-Info AVP . . . . .	26
4.2.9.	Reply-Message AVP . . . . .	27
4.3.	NAS Authentication AVPs . . . . .	27
4.3.1.	User-Password AVP . . . . .	28
4.3.2.	Password-Retry AVP . . . . .	28
4.3.3.	Prompt AVP . . . . .	28
4.3.4.	CHAP-Auth AVP . . . . .	29
4.3.5.	CHAP-Algorithm AVP . . . . .	29
4.3.6.	CHAP-Ident AVP . . . . .	29
4.3.7.	CHAP-Response AVP . . . . .	29

4.3.8.	CHAP-Challenge AVP . . . . .	29
4.3.9.	ARAP-Password AVP . . . . .	30
4.3.10.	ARAP-Challenge-Response AVP . . . . .	30
4.3.11.	ARAP-Security AVP . . . . .	30
4.3.12.	ARAP-Security-Data AVP . . . . .	30
4.4.	NAS Authorization AVPs . . . . .	30
4.4.1.	Service-Type AVP . . . . .	32
4.4.2.	Callback-Number AVP . . . . .	33
4.4.3.	Callback-Id AVP . . . . .	33
4.4.4.	Idle-Timeout AVP . . . . .	33
4.4.5.	Port-Limit AVP . . . . .	33
4.4.6.	NAS-Filter-Rule AVP . . . . .	33
4.4.7.	Filter-Id AVP . . . . .	34
4.4.8.	Configuration-Token AVP . . . . .	34
4.4.9.	QoS-Filter-Rule AVP . . . . .	34
4.4.10.	Framed Access Authorization AVPs . . . . .	35
4.4.10.1.	Framed-Protocol AVP . . . . .	35
4.4.10.2.	Framed-Routing AVP . . . . .	35
4.4.10.3.	Framed-MTU AVP . . . . .	36
4.4.10.4.	Framed-Compression AVP . . . . .	36
4.4.10.5.	IP Access Authorization AVPs . . . . .	36
4.4.10.5.1.	Framed-IP-Address AVP . . . . .	36
4.4.10.5.2.	Framed-IP-Netmask AVP . . . . .	36
4.4.10.5.3.	Framed-Route AVP . . . . .	37
4.4.10.5.4.	Framed-Pool AVP . . . . .	37
4.4.10.5.5.	Framed-Interface-Id AVP . . . . .	37
4.4.10.5.6.	Framed-IPv6-Prefix AVP . . . . .	38
4.4.10.5.7.	Framed-IPv6-Route AVP . . . . .	38
4.4.10.5.8.	Framed-IPv6-Pool AVP . . . . .	38
4.4.10.6.	IPX Access AVPs . . . . .	38
4.4.10.6.1.	Framed-IPX-Network AVP . . . . .	38
4.4.10.7.	AppleTalk Network Access AVPs . . . . .	39
4.4.10.7.1.	Framed-AppleTalk-Link AVP . . . . .	39
4.4.10.7.2.	Framed-AppleTalk-Network AVP . . . . .	39
4.4.10.7.3.	Framed-AppleTalk-Zone AVP . . . . .	39
4.4.10.8.	AppleTalk Remote Access AVPs . . . . .	40
4.4.10.8.1.	ARAP-Features AVP . . . . .	40
4.4.10.8.2.	ARAP-Zone-Access AVP . . . . .	40
4.4.11.	Non-Framed Access Authorization AVPs . . . . .	40
4.4.11.1.	Login-IP-Host AVP . . . . .	40
4.4.11.2.	Login-IPv6-Host AVP . . . . .	41
4.4.11.3.	Login-Service AVP . . . . .	41
4.4.11.4.	TCP Services . . . . .	41
4.4.11.4.1.	Login-TCP-Port AVP . . . . .	41
4.4.11.5.	LAT Services . . . . .	41
4.4.11.5.1.	Login-LAT-Service AVP . . . . .	41
4.4.11.5.2.	Login-LAT-Node AVP . . . . .	42
4.4.11.5.3.	Login-LAT-Group AVP . . . . .	43

4.4.11.5.4. Login-LAT-Port AVP . . . . .	43
4.5. NAS Tunneling AVPs . . . . .	43
4.5.1. Tunneling AVP . . . . .	44
4.5.2. Tunnel-Type AVP . . . . .	44
4.5.3. Tunnel-Medium-Type AVP . . . . .	45
4.5.4. Tunnel-Client-Endpoint AVP . . . . .	45
4.5.5. Tunnel-Server-Endpoint AVP . . . . .	46
4.5.6. Tunnel-Password AVP . . . . .	47
4.5.7. Tunnel-Private-Group-Id AVP . . . . .	47
4.5.8. Tunnel-Assignment-Id AVP . . . . .	47
4.5.9. Tunnel-Preference AVP . . . . .	48
4.5.10. Tunnel-Client-Auth-Id AVP . . . . .	49
4.5.11. Tunnel-Server-Auth-Id AVP . . . . .	49
4.6. NAS Accounting AVPs . . . . .	49
4.6.1. Accounting-Input-Octets AVP . . . . .	50
4.6.2. Accounting-Output-Octets AVP . . . . .	51
4.6.3. Accounting-Input-Packets AVP . . . . .	51
4.6.4. Accounting-Output-Packets AVP . . . . .	51
4.6.5. Acct-Session-Time AVP . . . . .	51
4.6.6. Acct-Authentic AVP . . . . .	51
4.6.7. Accounting-Auth-Method AVP . . . . .	52
4.6.8. Acct-Delay-Time AVP . . . . .	52
4.6.9. Acct-Link-Count AVP . . . . .	52
4.6.10. Acct-Tunnel-Connection AVP . . . . .	53
4.6.11. Acct-Tunnel-Packets-Lost AVP . . . . .	53
5. AVP Occurrence Tables . . . . .	53
5.1. AA-Request/Answer AVP Table . . . . .	54
5.2. Accounting AVP Tables . . . . .	56
5.2.1. Framed Access Accounting AVP Table . . . . .	56
5.2.2. Non-Framed Access Accounting AVP Table . . . . .	58
6. Unicode Considerations . . . . .	60
7. IANA Considerations . . . . .	60
8. Security Considerations . . . . .	61
8.1. Authentication Considerations . . . . .	61
8.2. AVP Considerations . . . . .	62
9. References . . . . .	62
9.1. Normative References . . . . .	62
9.2. Informative References . . . . .	63
Appendix A. Acknowledgements . . . . .	66
A.1. This Document . . . . .	66
A.2. RFC 4005 . . . . .	66

## 1. Introduction

This document describes the Diameter protocol application used for AAA in the Network Access Server (NAS) environment. When combined with the Diameter Base protocol [RFC6733], Transport Profile [RFC3539], and EAP [RFC4072] specifications, this specification

satisfies the NAS-related requirements defined in Aboba, et al. [RFC2989] and Beadles & Mitton [RFC3169].

First, this document describes the operation of a Diameter NAS application. Then it defines the Diameter message Command-Codes. The following sections list the AVPs used in these messages, grouped by common usage. These are session identification, authentication, authorization, tunneling, and accounting. The authorization AVPs are further broken down by service type.

#### 1.1. Changes from RFC 4005

This document obsoletes RFC 4005 and is not backward compatible with that document. An overview of some of the major changes is given below.

- o All of the material regarding RADIUS/Diameter protocol interactions has been removed; however, where AVPs are derived from RADIUS Attributes, the range and format of those Attribute values have been retained for ease of transition.
- o The Command Code Format (CCF) [RFC6733] for the Accounting-Request and Accounting-Answer messages has been changed to explicitly require the inclusion of the Acct-Application-Id AVP and exclude the Vendor-Specific-Application-Id AVP. Normally, this type of change would require the allocation of a new command code and consequently, a new application-id (See Section 1.3.3 of [RFC6733]). However, the presence of an instance of the Acct-Application-Id AVP was required in RFC 4005, as well:

The ACR message [BASE] is sent by the NAS to report its session information to a target server downstream.

Either of Acct-Application-Id or Vendor-Specific-Application-Id AVPs MUST be present. If the Vendor-Specific-Application-Id grouped AVP is present, it must have an Acct-Application-Id inside.

Thus, though the syntax of the commands has changed, the semantics have not (with the caveat that the Acct-Application-Id AVP can no longer be contained in the Vendor-Specific-Application-Id AVP).

- o The lists of RADIUS attribute values have been deleted in favor of references to the appropriate IANA registries.
- o The accounting model to be used is now specified (see Section 1.6).

There are many other miscellaneous fixes that have been introduced in this document that may not be considered significant but they are useful nonetheless. Examples are fixes to example IP addresses, addition of clarifying references, etc. All of the errata previously filed against RFC 4005 have been fixed. A comprehensive list of changes is not shown here for practical reasons.

## 1.2. Terminology

Section 1.2 of the Diameter base protocol specification [RFC6733] defines most of the terminology used in this document. Additionally, the following terms and acronyms are used in this application:

NAS (Network Access Server)

A device that provides an access service for a user to a network. The service may be a network connection or a value-added service such as terminal emulation [RFC2881].

PPP (Point-to-Point Protocol)

A multiprotocol serial datalink. PPP is the primary IP datalink used for dial-in NAS connection service [RFC1661].

CHAP (Challenge Handshake Authentication Protocol)

An authentication process used in PPP [RFC1994].

PAP (Password Authentication Protocol)

A deprecated PPP authentication process, but often used for backward compatibility [RFC1334].

SLIP (Serial Line Interface Protocol)

A serial datalink that only supports IP. A design prior to PPP.

ARAP (Appletalk Remote Access Protocol)

A serial datalink for accessing Appletalk networks [ARAP].

IPX (Internet Packet Exchange)

The network protocol used by NetWare networks [IPX].

L2TP (Layer Two Tunneling Protocol)

L2TP [RFC3931] provides a dynamic mechanism for tunneling Layer 2 "circuits" across a packet-oriented data network.

LAC (L2TP Access Concentrator)

An L2TP Control Connection Endpoint being used to cross-connect an L2TP session directly to a data link [RFC3931].

LAT (Local Area Transport)

A Digital Equipment Corp. LAN protocol for terminal services [LAT].

LCP (Link Control Protocol)

One of the three major components of PPP [RFC1661]. LCP is used to automatically agree upon encapsulation format options, handle varying limits on sizes of packets, detect a looped-back link and other common misconfiguration errors, and terminate the link. Other optional facilities provided are authentication of the identity of its peer on the link, and determination when a link is functioning properly and when it is failing.

PPTP (Point-to-Point Tunneling Protocol)

A protocol which allows PPP to be tunneled through an IP network [RFC2637].

VPN (Virtual Private Network)

In this document, this term is used to describe access services that use tunneling methods.

### 1.3. Requirements Language

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

The use of "MUST" and "MUST NOT" in the AVP Flag rules columns of AVP Tables in this document refers to AVP flags ([RFC6733], Section 4.1) that:

- o MUST be set to 1 in the AVP Header ("MUST" column) and
- o MUST NOT be set to 1 ("MUST NOT" column)



#### 1.4. Advertising Application Support

Diameter nodes conforming to this specification MUST advertise support by including the value of one (1) in the Auth-Application-Id of the Capabilities-Exchange-Request (CER) message [RFC6733].

#### 1.5. Application Identification

When used in this application, the Auth-Application-Id AVP MUST be set to the value one (1) in the following messages

- o AA-Request (Section 3.1)
- o Re-Auth-Request (Section 3.3)
- o Session-Termination-Request (Section 3.5)
- o Abort-Session-Request (Section 3.7)

#### 1.6. Accounting Model

It is RECOMMENDED that the coupled accounting model (RFC 6733, Section 9.3) be used with this application; therefore, the value of the Acct-Application-Id AVP in the Accounting-Request (Section 3.10) and Accounting-Answer (Section 3.9) messages SHOULD be set to one (1).

### 2. NAS Calls, Ports, and Sessions

The arrival of a new call or service connection at a port of a Network Access Server (NAS) starts a Diameter NAS Application message exchange. Information about the call, the identity of the user, and the user's authentication information are packaged into a Diameter AA-Request (AAR) message and sent to a server.

The server processes the information and responds with a Diameter AA-Answer (AAA) message that contains authorization information for the NAS, or a failure code (Result-Code AVP). A value of DIAMETER\_MULTI\_ROUND\_AUTH indicates an additional authentication exchange, and several AAR and AAA messages may be exchanged until the transaction completes.

#### 2.1. Diameter Session Establishment

When the authentication or authorization exchange completes successfully, the NAS application SHOULD start a session context. If the Result-Code of DIAMETER\_MULTI\_ROUND\_AUTH is returned, the exchange continues until a success or error is returned.

If accounting is active, the application MUST also send an Accounting message [RFC6733]. An Accounting-Record-Type of START\_RECORD is sent for a new session. If a session fails to start, the EVENT\_RECORD message is sent with the reason for the failure described.

Note that the return of an unsupportable Accounting-Realtime-Required value [RFC6733] would result in a failure to establish the session.

## 2.2. Diameter Session Reauthentication or Reauthorization

The Diameter Base protocol allows users to be periodically reauthenticated and/or reauthorized. In such instances, the Session-Id AVP in the AAR message MUST be the same as the one present in the original authentication/authorization message.

A Diameter server informs the NAS of the maximum time allowed before reauthentication or reauthorization via the Authorization-Lifetime AVP [RFC6733]. A NAS MAY reauthenticate and/or reauthorize before the end, but A NAS MUST reauthenticate and/or reauthorize at the end of the period provided by the Authorization-Lifetime AVP. The failure of a reauthentication exchange will terminate the service.

Furthermore, it is possible for Diameter servers to issue an unsolicited reauthentication and/or reauthorization request (e.g., Re-Auth-Request (RAR) message [RFC6733]) to the NAS. Upon receipt of such a message, the NAS MUST respond to the request with a Re-Auth-Answer (RAA) message [RFC6733].

If the RAR properly identifies an active session, the NAS will initiate a new local reauthentication or authorization sequence as indicated by the Re-Auth-Request-Type value. This will cause the NAS to send a new AAR message using the existing Session-Id. The server will respond with an AAA message to specify the new service parameters.

If accounting is active, every change of authentication or authorization SHOULD generate an accounting message. If the NAS service is a continuation of the prior user context, then an Accounting-Record-Type of INTERIM\_RECORD indicating the new session attributes and cumulative status would be appropriate. If a new user or a significant change in authorization is detected by the NAS, then the service may send two messages of the types STOP\_RECORD and START\_RECORD. Accounting may change the subsession identifiers (Acct-Session-ID, or Acct-Sub-Session-Id) to indicate such subsessions. A service may also use a different Session-Id value for accounting (see Section 9.6 of [RFC6733]).

However, the Diameter Session-ID AVP value used for the initial authorization exchange MUST be used to generate an STR message when the session context is terminated.

### 2.3. Diameter Session Termination

When a NAS receives an indication that a user's session is being disconnected by the client (e.g., an LCP Terminate-Request message [RFC1661] is received) or an administrative command, the NAS MUST issue a Session-Termination-Request (STR) [RFC6733] to its Diameter Server. This will ensure that any resources maintained on the servers are freed appropriately.

Furthermore, a NAS that receives an Abort-Session-Request (ASR) [RFC6733] MUST issue an Abort-Session-Answer (ASA) if the session identified is active and disconnect the PPP (or tunneling) session.

If accounting is active, an Accounting STOP\_RECORD message [RFC6733] MUST be sent upon termination of the session context.

More information on Diameter Session Termination can be found in Sections 8.4 and 8.5 of [RFC6733].

### 3. Diameter NAS Application Messages

This section defines the Diameter message Command-Code [RFC6733] values that MUST be supported by all Diameter implementations conforming to this specification. The Command Codes are as follows:

Command Name	Abbrev.	Code	Reference
AA-Request	AAR	265	Section 3.1
AA-Answer	AAA	265	Section 3.2
Re-Auth-Request	RAR	258	Section 3.3
Re-Auth-Answer	RAA	258	Section 3.4
Session-Termination-Request	STR	275	Section 3.5
Session-Termination-Answer	STA	275	Section 3.6
Abort-Session-Request	ASR	274	Section 3.7
Abort-Session-Answer	ASA	274	Section 3.8
Accounting-Request	ACR	271	Section 3.9
Accounting-Answer	ACA	271	Section 3.10

Note that the message formats in the following sub-sections use the standard Diameter Command Code Format ([RFC6733], Section 3.2).

### 3.1. AA-Request (AAR) Command

The AA-Request (AAR), which is indicated by setting the Command-Code field to 265 and the 'R' bit in the Command Flags field, is used to request authentication and/or authorization for a given NAS user. The type of request is identified through the Auth-Request-Type AVP [RFC6733]. The recommended value for most situations is `AUTHORIZE_AUTHENTICATE`.

If Authentication is requested, the User-Name attribute SHOULD be present, as well as any additional authentication AVPs that would carry the password information. A request for authorization SHOULD only include the information from which the authorization will be performed, such as the User-Name, Called-Station-Id, or Calling-Station-Id AVPs. All requests SHOULD contain AVPs uniquely identifying the source of the call, such as Origin-Host and NAS-Port. Certain networks MAY use different AVPs for authorization purposes. A request for authorization will include some AVPs defined in Section 4.4.

It is possible for a single session to be authorized first and then for an authentication request to follow.

This AA-Request message MAY be the result of a multi-round authentication exchange, which occurs when the AA-Answer message is received with the Result-Code AVP set to `DIAMETER_MULTI_ROUND_AUTH`. A subsequent AAR message SHOULD be sent, with the User-Password AVP that includes the user's response to the prompt, and MUST include any State AVPs that were present in the AAA message.

#### Message Format

```
<AA-Request> ::= < Diameter Header: 265, REQ, PXY >
                  < Session-Id >
                  { Auth-Application-Id }
                  { Origin-Host }
                  { Origin-Realm }
                  { Destination-Realm }
                  { Auth-Request-Type }
                  [ Destination-Host ]
                  [ NAS-Identifier ]
                  [ NAS-IP-Address ]
                  [ NAS-IPv6-Address ]
                  [ NAS-Port ]
                  [ NAS-Port-Id ]
                  [ NAS-Port-Type ]
                  [ Origin-AAA-Protocol ]
                  [ Origin-State-Id ]
```

```
[ Port-Limit ]
[ User-Name ]
[ User-Password ]
[ Service-Type ]
[ State ]
[ Authorization-Lifetime ]
[ Auth-Grace-Period ]
[ Auth-Session-State ]
[ Callback-Number ]
[ Called-Station-Id ]
[ Calling-Station-Id ]
[ Originating-Line-Info ]
[ Connect-Info ]
[ CHAP-Auth ]
[ CHAP-Challenge ]
* [ Framed-Compression ]
[ Framed-Interface-Id ]
[ Framed-IP-Address ]
* [ Framed-IPv6-Prefix ]
[ Framed-IP-Netmask ]
[ Framed-MTU ]
[ Framed-Protocol ]
[ ARAP-Password ]
[ ARAP-Security ]
* [ ARAP-Security-Data ]
* [ Login-IP-Host ]
* [ Login-IPv6-Host ]
[ Login-LAT-Group ]
[ Login-LAT-Node ]
[ Login-LAT-Port ]
[ Login-LAT-Service ]
* [ Tunneling ]
* [ Proxy-Info ]
* [ Route-Record ]
* [ AVP ]
```

Figure 1

### 3.2. AA-Answer (AAA) Command

The AA-Answer (AAA) message is indicated by setting the Command-Code field to 265 and clearing the 'R' bit in the Command Flags field. It is sent in response to the AA-Request (AAR) message. If authorization was requested, a successful response will include the authorization AVPs appropriate for the service being provided, as defined in Section 4.4.

For authentication exchanges requiring more than a single round trip, the server MUST set the Result-Code AVP to DIAMETER\_MULTI\_ROUND\_AUTH. An AAA message with this result code MAY include one Reply-Message or more and MAY include zero or one State AVPs.

If the Reply-Message AVP was present, the network access server SHOULD send the text to the user's client to display to the user, instructing the client to prompt the user for a response. For example, this can be achieved in PPP via PAP. If it is impossible to deliver the text prompt to the user, the Diameter NAS Application client MUST treat the AA-Answer (AAA) with the Reply-Message AVP as an error and deny access.

#### Message Format

```
<AA-Answer> ::= < Diameter Header: 265, PXY >
               < Session-Id >
               { Auth-Application-Id }
               { Auth-Request-Type }
               { Result-Code }
               { Origin-Host }
               { Origin-Realm }
               [ User-Name ]
               [ Service-Type ]
               * [ Class ]
               * [ Configuration-Token ]
               [ Acct-Interim-Interval ]
               [ Error-Message ]
               [ Error-Reporting-Host ]
               * [ Failed-AVP ]
               [ Idle-Timeout ]
               [ Authorization-Lifetime ]
               [ Auth-Grace-Period ]
               [ Auth-Session-State ]
               [ Re-Auth-Request-Type ]
               [ Multi-Round-Time-Out ]
               [ Session-Timeout ]
               [ State ]
               * [ Reply-Message ]
               [ Origin-AAA-Protocol ]
               [ Origin-State-Id ]
               * [ Filter-Id ]
               [ Password-Retry ]
               [ Port-Limit ]
               [ Prompt ]
               [ ARAP-Challenge-Response ]
               [ ARAP-Features ]
               [ ARAP-Security ]
```

```
* [ ARAP-Security-Data ]
  [ ARAP-Zone-Access ]
  [ Callback-Id ]
  [ Callback-Number ]
  [ Framed-Appletalk-Link ]
* [ Framed-Appletalk-Network ]
  [ Framed-Appletalk-Zone ]
* [ Framed-Compression ]
  [ Framed-Interface-Id ]
  [ Framed-IP-Address ]
* [ Framed-IPv6-Prefix ]
  [ Framed-IPv6-Pool ]
* [ Framed-IPv6-Route ]
  [ Framed-IP-Netmask ]
* [ Framed-Route ]
  [ Framed-Pool ]
  [ Framed-IPX-Network ]
  [ Framed-MTU ]
  [ Framed-Protocol ]
  [ Framed-Routing ]
* [ Login-IP-Host ]
* [ Login-IPv6-Host ]
  [ Login-LAT-Group ]
  [ Login-LAT-Node ]
  [ Login-LAT-Port ]
  [ Login-LAT-Service ]
  [ Login-Service ]
  [ Login-TCP-Port ]
* [ NAS-Filter-Rule ]
* [ QoS-Filter-Rule ]
* [ Tunneling ]
* [ Redirect-Host ]
  [ Redirect-Host-Usage ]
  [ Redirect-Max-Cache-Time ]
* [ Proxy-Info ]
* [ AVP ]
```

Figure 2

### 3.3. Re-Auth-Request (RAR) Command

A Diameter server can initiate re-authentication and/or re-authorization for a particular session by issuing a Re-Auth-Request (RAR) message [RFC6733].

For example, for pre-paid services, the Diameter server that originally authorized a session may need some confirmation that the user is still using the services.

If a NAS receives an RAR message with Session-Id equal to a currently active session and a Re-Auth-Type that includes authentication, it MUST initiate a re-authentication toward the user, if the service supports this particular feature.

#### Message Format

```
<RA-Request> ::= < 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-AAA-Protocol ]
    [ Origin-State-Id ]
    [ NAS-Identifier ]
    [ NAS-IP-Address ]
    [ NAS-IPv6-Address ]
    [ NAS-Port ]
    [ NAS-Port-Id ]
    [ NAS-Port-Type ]
    [ Service-Type ]
    [ Framed-IP-Address ]
    [ Framed-IPv6-Prefix ]
    [ Framed-Interface-Id ]
    [ Called-Station-Id ]
    [ Calling-Station-Id ]
    [ Originating-Line-Info ]
    [ Acct-Session-Id ]
    [ Acct-Multi-Session-Id ]
    [ State ]
    * [ Class ]
    [ Reply-Message ]
    * [ Proxy-Info ]
    * [ Route-Record ]
    * [ AVP ]
```

Figure 3

#### 3.4. Re-Auth-Answer (RAA) Command

The Re-Auth-Answer (RAA) message [RFC6733] is sent in response to the RAR. The Result-Code AVP MUST be present and indicates the disposition of the request.



A successful RAA transaction MUST be followed by an AAR message.

#### Message Format

```
<RA-Answer> ::= < Diameter Header: 258, PXY >
               < Session-Id >
               { Result-Code }
               { Origin-Host }
               { Origin-Realm }
               [ User-Name ]
               [ Origin-AAA-Protocol ]
               [ Origin-State-Id ]
               [ Error-Message ]
               [ Error-Reporting-Host ]
               * [ Failed-AVP ]
               * [ Redirected-Host ]
               [ Redirected-Host-Usage ]
               [ Redirected-Host-Cache-Time ]
               [ Service-Type ]
               * [ Configuration-Token ]
               [ Idle-Timeout ]
               [ Authorization-Lifetime ]
               [ Auth-Grace-Period ]
               [ Re-Auth-Request-Type ]
               [ State ]
               * [ Class ]
               * [ Reply-Message ]
               [ Prompt ]
               * [ Proxy-Info ]
               * [ AVP ]
```

Figure 4

### 3.5. Session-Termination-Request (STR) Command

The Session-Termination-Request (STR) message [RFC6733] is sent by the NAS to inform the Diameter Server that an authenticated and/or authorized session is being terminated.

#### Message Format

```
<ST-Request> ::= < Diameter Header: 275, REQ, PXY >
               < Session-Id >
               { Origin-Host }
               { Origin-Realm }
               { Destination-Realm }
               { Auth-Application-Id }
               { Termination-Cause }
```

```
[ User-Name ]
[ Destination-Host ]
* [ Class ]
[ Origin-AAA-Protocol ]
[ Origin-State-Id ]
* [ Proxy-Info ]
* [ Route-Record ]
* [ AVP ]
```

Figure 5

### 3.6. Session-Termination-Answer (STA) Command

The Session-Termination-Answer (STA) message [RFC6733] is sent by the Diameter Server to acknowledge the notification that the session has been terminated. The Result-Code AVP MUST be present and MAY contain an indication that an error occurred while the STR was being serviced.

Upon sending the STA, the Diameter Server MUST release all resources for the session indicated by the Session-Id AVP. Any intermediate server in the Proxy-Chain MAY also release any resources, if necessary.

#### Message Format

```
<ST-Answer> ::= < Diameter Header: 275, PXY >
               < Session-Id >
               { Result-Code }
               { Origin-Host }
               { Origin-Realm }
               [ User-Name ]
               * [ Class ]
               [ Error-Message ]
               [ Error-Reporting-Host ]
               * [ Failed-AVP ]
               [ Origin-AAA-Protocol ]
               [ Origin-State-Id ]
               * [ Redirect-Host ]
               [ Redirect-Host-Usase ]
               [ Redirect-Max-Cache-Time ]
               * [ Proxy-Info ]
               * [ AVP ]
```

Figure 6

### 3.7. Abort-Session-Request (ASR) Command

The Abort-Session-Request (ASR) message [RFC6733] can be sent by any Diameter server to the NAS providing session service to request that the session identified by the Session-Id be stopped.

#### Message Format

```

<AS-Request> ::= < Diameter Header: 274, REQ, PXY >
    < Session-Id >
    { Origin-Host }
    { Origin-Realm }
    { Destination-Realm }
    { Destination-Host }
    { Auth-Application-Id }
    [ User-Name ]
    [ Origin-AAA-Protocol ]
    [ Origin-State-Id ]
    [ NAS-Identifier ]
    [ NAS-IP-Address ]
    [ NAS-IPv6-Address ]
    [ NAS-Port ]
    [ NAS-Port-Id ]
    [ NAS-Port-Type ]
    [ Service-Type ]
    [ Framed-IP-Address ]
    [ Framed-IPv6-Prefix ]
    [ Framed-Interface-Id ]
    [ Called-Station-Id ]
    [ Calling-Station-Id ]
    [ Originating-Line-Info ]
    [ Acct-Session-Id ]
    [ Acct-Multi-Session-Id ]
    [ State ]
    * [ Class ]
    * [ Reply-Message ]
    * [ Proxy-Info ]
    * [ Route-Record ]
    * [ AVP ]

```

Figure 7

### 3.8. Abort-Session-Answer (ASA) Command

The ASA message [RFC6733] is sent in response to the ASR. The Result-Code AVP MUST be present and indicates the disposition of the request.

If the session identified by Session-Id in the ASR was successfully terminated, Result-Code is set to DIAMETER\_SUCCESS. If the session

is not currently active, the Result-Code AVP is set to DIAMETER\_UNKNOWN\_SESSION\_ID. If the access device does not stop the session for any other reason, the Result-Code AVP is set to DIAMETER\_UNABLE\_TO\_COMPLY.

#### Message Format

```
<AS-Answer> ::= < Diameter Header: 274, PXY >
               < Session-Id >
               { Result-Code }
               { Origin-Host }
               { Origin-Realm }
               [ User-Name ]
               [ Origin-AAA-Protocol ]
               [ Origin-State-Id ]
               [ State ]
               [ Error-Message ]
               [ Error-Reporting-Host ]
               * [ Failed-AVP ]
               * [ Redirected-Host ]
               [ Redirected-Host-Usage ]
               [ Redirected-Max-Cache-Time ]
               * [ Proxy-Info ]
               * [ AVP ]
```

Figure 8

### 3.9. Accounting-Request (ACR) Command

The ACR message [RFC6733] is sent by the NAS to report its session information to a target server downstream.

The Acct-Application-Id AVP MUST be present.

The AVPs listed in the Base protocol specification [RFC6733] MUST be assumed to be present, as appropriate. NAS service-specific accounting AVPs SHOULD be present as described in Section 4.6 and the rest of this specification.

#### Message Format

```
<AC-Request> ::= < Diameter Header: 271, REQ, PXY >
               < Session-Id >
               { Origin-Host }
               { Origin-Realm }
               { Destination-Realm }
               { Accounting-Record-Type }
               { Accounting-Record-Number }
```

```
{ Acct-Application-Id }
[ User-Name ]
[ Accounting-Sub-Session-Id ]
[ Acct-Session-Id ]
[ Acct-Multi-Session-Id ]
[ Origin-AAA-Protocol ]
[ Origin-State-Id ]
[ Destination-Host ]
[ Event-Timestamp ]
[ Acct-Delay-Time ]
[ NAS-Identifier ]
[ NAS-IP-Address ]
[ NAS-IPv6-Address ]
[ NAS-Port ]
[ NAS-Port-Id ]
[ NAS-Port-Type ]
* [ Class ]
[ Service-Type ]
[ Termination-Cause ]
[ Accounting-Input-Octets ]
[ Accounting-Input-Packets ]
[ Accounting-Output-Octets ]
[ Accounting-Output-Packets ]
[ Acct-Authentic ]
[ Accounting-Auth-Method ]
[ Acct-Link-Count ]
[ Acct-Session-Time ]
[ Acct-Tunnel-Connection ]
[ Acct-Tunnel-Packets-Lost ]
[ Callback-Id ]
[ Callback-Number ]
[ Called-Station-Id ]
[ Calling-Station-Id ]
* [ Connection-Info ]
[ Originating-Line-Info ]
[ Authorization-Lifetime ]
[ Session-Timeout ]
[ Idle-Timeout ]
[ Port-Limit ]
[ Accounting-Realtime-Required ]
[ Acct-Interim-Interval ]
* [ Filter-Id ]
* [ NAS-Filter-Rule ]
* [ QoS-Filter-Rule ]
[ Framed-AppleTalk-Link ]
[ Framed-AppleTalk-Network ]
[ Framed-AppleTalk-Zone ]
[ Framed-Compression ]
```

```
[ Framed-Interface-Id ]
[ Framed-IP-Address ]
[ Framed-IP-Netmask ]
* [ Framed-IPv6-Prefix ]
[ Framed-IPv6-Pool ]
* [ Framed-IPv6-Route ]
[ Framed-IPX-Network ]
[ Framed-MTU ]
[ Framed-Pool ]
[ Framed-Protocol ]
* [ Framed-Route ]
[ Framed-Routing ]
* [ Login-IP-Host ]
* [ Login-IPv6-Host ]
[ Login-LAT-Group ]
[ Login-LAT-Node ]
[ Login-LAT-Port ]
[ Login-LAT-Service ]
[ Login-Service ]
[ Login-TCP-Port ]
* [ Tunneling ]
* [ Proxy-Info ]
* [ Route-Record ]
* [ AVP ]
```

Figure 9

### 3.10. Accounting-Answer (ACA) Command

The ACA message [RFC6733] is used to acknowledge an Accounting-Request command. The Accounting-Answer command contains the same Session-Id as the Request.

Only the target Diameter Server or home Diameter Server SHOULD respond with the Accounting-Answer command.

The Acct-Application-Id AVP MUST be present.

The AVPs listed in the Base protocol specification [RFC6733] MUST be assumed to be present, as appropriate. NAS service-specific accounting AVPs SHOULD be present as described in Section 4.6 and the rest of this specification.

#### Message Format

```
<AC-Answer> ::= < Diameter Header: 271, PXY >
               < Session-Id >
               { Result-Code }
```

```
{ Origin-Host }
{ Origin-Realm }
{ Accounting-Record-Type }
{ Accounting-Record-Number }
{ Acct-Application-Id }
[ User-Name ]
[ Accounting-Sub-Session-Id ]
[ Acct-Session-Id ]
[ Acct-Multi-Session-Id ]
[ Event-Timestamp ]
[ Error-Message ]
[ Error-Reporting-Host ]
* [ Failed-AVP ]
[ Origin-AAA-Protocol ]
[ Origin-State-Id ]
[ NAS-Identifier ]
[ NAS-IP-Address ]
[ NAS-IPv6-Address ]
[ NAS-Port ]
[ NAS-Port-Id ]
[ NAS-Port-Type ]
[ Service-Type ]
[ Termination-Cause ]
[ Accounting-Realtime-Required ]
[ Acct-Interim-Interval ]
* [ Class ]
* [ Proxy-Info ]
* [ AVP ]
```

Figure 10

#### 4. Diameter NAS Application AVPs

The following sections define a new derived AVP data format, a set of application-specific AVPs and describe the use of AVPs defined in other documents by the Diameter NAS Application.

##### 4.1. Derived AVP Data Formats

###### 4.1.1. QoSFilterRule

The QoSFilterRule format is derived from the OctetString AVP Base Format. It uses the ASCII charset. Packets may be marked or metered based on the following information:

- o Direction (in or out)
- o Source and destination IP address (possibly masked)

- o Protocol
- o Source and destination port (lists or ranges)
- o DSCP values (no mask or range)

Rules for the appropriate direction are evaluated in order; the first matched rule terminates the evaluation. Each packet is evaluated once. If no rule matches, the packet is treated as best effort. An access device unable to interpret or apply a QoS rule SHOULD NOT terminate the session.

QoSFilterRule filters MUST follow the following format:

```
action dir proto from src to dst [options]
```

where

action

tag Mark packet with a specific DSCP [RFC2474]

meter Meter traffic

dir The format is as described under IPFilterRule [RFC6733]

proto The format is as described under IPFilterRule [RFC6733]

src and dst The format is as described under IPFilterRule [RFC6733]

The options are described in Section 4.4.9.

The rule syntax is a modified subset of ipfw(8) from FreeBSD, and the ipfw.c code may provide a useful base for implementations.

#### 4.2. NAS Session AVPs

Diameter reserves the AVP Codes 0 - 255 for RADIUS Attributes that are implemented in Diameter.



#### 4.2.1. Call and Session Information

This section describes the AVPs specific to Diameter applications that are needed to identify the call and session context and status information. On a request, this information allows the server to qualify the session.

These AVPs are used in addition to the following AVPs from the base protocol specification [RFC6733]:

Session-Id  
Auth-Application-Id  
Origin-Host  
Origin-Realm  
Auth-Request-Type  
Termination-Cause

The following table gives the possible flag values for the session level AVPs.

		+-----+   AVP Flag     Rules   +-----+	
		MUST	MUST NOT
Attribute Name	Section Defined		
NAS-Port	4.2.2	M	V
NAS-Port-Id	4.2.3	M	V
NAS-Port-Type	4.2.4	M	V
Called-Station-Id	4.2.5	M	V
Calling-Station-Id	4.2.6	M	V
Connect-Info	4.2.7	M	V
Originating-Line-Info	4.2.8	M	V
Reply-Message	4.2.9	M	V

#### 4.2.2. NAS-Port AVP

The NAS-Port AVP (AVP Code 5) is of type Unsigned32 and contains the physical or virtual port number of the NAS which is authenticating the user. Note that "port" is meant in its sense as a service connection on the NAS, not as an IP protocol identifier, and hence the format and contents of the string that identifies the port are specific to the NAS implementation.

Either the NAS-Port AVP or the NAS-Port-Id AVP (Section 4.2.3) SHOULD be present in the AA-Request (AAR, Section 3.1) command if the NAS differentiates among its ports.

#### 4.2.3. NAS-Port-Id AVP

The NAS-Port-Id AVP (AVP Code 87) is of type UTF8String and consists of 7-bit ASCII text identifying the port of the NAS authenticating the user. Note that "port" is meant in its sense as a service connection on the NAS, not as an IP protocol identifier.

Either the NAS-Port-Id AVP or the NAS-Port AVP (Section 4.2.2) SHOULD be present in the AA-Request (AAR, Section 3.1) command if the NAS differentiates among its ports. NAS-Port-Id is intended for use by NASes that cannot conveniently number their ports.

#### 4.2.4. NAS-Port-Type AVP

The NAS-Port-Type AVP (AVP Code 61) is of type Enumerated and contains the type of the port on which the NAS is authenticating the user. This AVP SHOULD be present if the NAS uses the same NAS-Port number ranges for different service types concurrently.

The currently supported values of the NAS-Port-Type AVP are listed in [RADIUSAttrVals].

#### 4.2.5. Called-Station-Id AVP

The Called-Station-Id AVP (AVP Code 30) is of type UTF8String contains a 7-bit ASCII string sent by the NAS to describe the Layer 2 address the user contacted in the request. For dialup access, this can be a phone number obtained by using the Dialed Number Identification Service (DNIS) or a similar technology. Note that this may be different from the phone number the call comes in on. For use with IEEE 802 access, the Called-Station-Id MAY contain a MAC address formatted as described in Congdon, et al. [RFC3580].

If the Called-Station-Id AVP is present in an AAR message, Auth-Request-Type AVP is set to AUTHORIZE\_ONLY and the User-Name AVP is absent, the Diameter Server MAY perform authorization based on this AVP. This can be used by a NAS to request whether a call should be answered based on the DNIS result.

Further codification of this field's allowed content and usage is outside the scope of this specification.

#### 4.2.6. Calling-Station-Id AVP

The Calling-Station-Id AVP (AVP Code 31) is of type UTF8String and contains a 7-bit ASCII string sent by the NAS to describe the Layer 2 address from which the user connected in the request. For dialup access, this is the phone number the call came from, using Automatic Number Identification (ANI) or a similar technology. For use with IEEE 802 access, the Calling-Station-Id AVP MAY contain a MAC address, formatted as described in RFC 3580.

If the Calling-Station-Id AVP is present in an AAR message, the Auth-Request-Type AVP is set to AUTHORIZE\_ONLY and the User-Name AVP is absent, the Diameter Server MAY perform authorization based on the value of this AVP. This can be used by a NAS to request whether a call should be answered based on the Layer 2 address (ANI, MAC Address, etc.)

Further codification of this field's allowed content and usage is outside the scope of this specification.

#### 4.2.7. Connect-Info AVP

The Connect-Info AVP (AVP Code 77) is of type UTF8String and is sent in the AA-Request message or an ACR message with the value of the Accounting-Record-Type AVP set to STOP. When sent in the AA-Request, it indicates the nature of the user's connection. The connection speed SHOULD be included at the beginning of the first Connect-Info AVP in the message. If the transmit and receive connection speeds differ, both may be included in the first AVP with the transmit speed listed first (the speed at which the NAS modem transmits), then a slash (/), then the receive speed, and then other optional information.

For example: "28800 V42BIS/LAPM" or "52000/31200 V90"

If sent in an ACR message with the value of the Accounting-Record-Type AVP set to STOP, this attribute may summarize statistics relating to session quality. For example, in IEEE 802.11, the Connect-Info AVP may contain information on the number of link layer retransmissions. The exact format of this attribute is implementation specific.

#### 4.2.8. Originating-Line-Info AVP

The Originating-Line-Info AVP (AVP Code 94) is of type OctetString and is sent by the NAS system to convey information about the origin of the call from an SS7 system.

The Originating Line Information (OLI) element indicates the nature and/or characteristics of the line from which a call originated

(e.g., pay phone, hotel, cellular). Telephone companies are starting to offer OLI to their customers as an option over Primary Rate Interface (PRI). Internet Service Providers (ISPs) can use OLI in addition to Called-Station-Id and Calling-Station-Id attributes to differentiate customer calls and to define different services.

The Value field contains two octets (00 - 99). ANSI T1.113 and BELLCORE 394 can be used for additional information about these values and their use. For information on the currently assigned values, see [ANITypes].

#### 4.2.9. Reply-Message AVP

The Reply-Message AVP (AVP Code 18) is of type UTF8String and contains text that MAY be displayed to the user. When used in an AA-Answer message with a successful Result-Code AVP, it indicates success. When found in an AAA message with a Result-Code other than DIAMETER\_SUCCESS, the AVP contains a failure message.

The Reply-Message AVP MAY contain text to prompt the user before another AA-Request attempt. When used in an AA-Answer message containing a Result-Code AVP with the value DIAMETER\_MULTI\_ROUND\_AUTH or in an Re-Auth-Request message, it MAY contain text to prompt the user for a response.

#### 4.3. NAS Authentication AVPs

This section defines the AVPs necessary to carry the authentication information in the Diameter protocol. The functionality defined here provides a RADIUS-like AAA service [RFC2865] over a more reliable and secure transport, as defined in the base protocol [RFC6733].

The following table gives the possible flag values for the session level AVPs.

Attribute Name	Section Defined	AVP Flag rules	
		MUST	MUST NOT
User-Password	4.3.1	M	V
Password-Retry	4.3.2	M	V
Prompt	4.3.3	M	V
CHAP-Auth	4.3.4	M	V
CHAP-Algorithm	4.3.5	M	V
CHAP-Ident	4.3.6	M	V

CHAP-Response	4.3.7	M	V
CHAP-Challenge	4.3.8	M	V
ARAP-Password	4.3.9	M	V
ARAP-Challenge-Response	4.3.10	M	V
ARAP-Security	4.3.11	M	V
ARAP-Security-Data	4.3.12	M	V
-----		-----+-----	

#### 4.3.1. User-Password AVP

The User-Password AVP (AVP Code 2) is of type OctetString and contains the password of the user to be authenticated, or the user's input in a multi-round authentication exchange.

The User-Password AVP contains a user password or one-time password and therefore represents sensitive information. As required by Fajardo, et al. [RFC6733], Diameter messages are encrypted by using IPsec [RFC4301] or TLS [RFC5246]. Unless this AVP is used for one-time passwords, the User-Password AVP SHOULD NOT be used in untrusted proxy environments without encrypting it by using end-to-end security techniques.

The clear-text password (prior to encryption) MUST NOT be longer than 128 bytes in length.

#### 4.3.2. Password-Retry AVP

The Password-Retry AVP (AVP Code 75) is of type Unsigned32 and MAY be included in the AA-Answer if the Result-Code indicates an authentication failure. The value of this AVP indicates how many authentication attempts a user is permitted before being disconnected. This AVP is primarily intended for use when the Framed-Protocol AVP (Section 4.4.10.1) is set to ARAP.

#### 4.3.3. Prompt AVP

The Prompt AVP (AVP Code 76) is of type Enumerated and MAY be present in the AA-Answer message. When present, it is used by the NAS to determine whether the user's response, when entered, should be echoed.

The supported values are listed in [RADIUSAttrVals]

#### 4.3.4. CHAP-Auth AVP

The CHAP-Auth AVP (AVP Code 402) is of type Grouped and contains the information necessary to authenticate a user using the PPP Challenge-Handshake Authentication Protocol (CHAP) [RFC1994]. If the CHAP-Auth AVP is found in a message, the CHAP-Challenge AVP (Section 4.3.8) MUST be present as well. The optional AVPs containing the CHAP response depend upon the value of the CHAP-Algorithm AVP (Section 4.3.8). The grouped AVP has the following ABNF grammar:

```
CHAP-Auth ::= < AVP Header: 402 >
              { CHAP-Algorithm }
              { CHAP-Ident }
              [ CHAP-Response ]
              * [ AVP ]
```

#### 4.3.5. CHAP-Algorithm AVP

The CHAP-Algorithm AVP (AVP Code 403) is of type Enumerated and contains the algorithm identifier used in the computation of the CHAP response [RFC1994]. The following values are currently supported:

CHAP with MD5                    5

The CHAP response is computed by using the procedure described in [RFC1994] This algorithm requires that the CHAP-Response AVP (Section 4.3.7) MUST be present in the CHAP-Auth AVP (Section 4.3.4).

#### 4.3.6. CHAP-Ident AVP

The CHAP-Ident AVP (AVP Code 404) is of type OctetString and contains the 1 octet CHAP Identifier used in the computation of the CHAP response [RFC1994]

#### 4.3.7. CHAP-Response AVP

The CHAP-Response AVP (AVP Code 405) is of type OctetString and contains the 16 octet authentication data provided by the user in response to the CHAP challenge [RFC1994].

#### 4.3.8. CHAP-Challenge AVP

The CHAP-Challenge AVP (AVP Code 60) is of type OctetString and contains the CHAP Challenge sent by the NAS to the CHAP peer [RFC1994].

#### 4.3.9. ARAP-Password AVP

The ARAP-Password AVP (AVP Code 70) is of type OctetString and is only present when the Framed-Protocol AVP (Section 4.4.10.1) is included in the message and is set to ARAP. This AVP MUST NOT be present if either the User-Password or the CHAP-Auth AVP is present. See Rigney, et al. [RFC2869] for more information on the contents of this AVP.

#### 4.3.10. ARAP-Challenge-Response AVP

The ARAP-Challenge-Response AVP (AVP Code 84) is of type OctetString and is only present when the Framed-Protocol AVP (Section 4.4.10.1) is included in the message and is set to ARAP. This AVP contains an 8 octet response to the dial-in client's challenge. The Diameter server calculates this value by taking the dial-in client's challenge from the high-order 8 octets of the ARAP-Password AVP and performing DES encryption on this value with the authenticating user's password as the key. If the user's password is fewer than 8 octets in length, the password is padded at the end with NULL octets to a length of 8 before it is used as a key.

#### 4.3.11. ARAP-Security AVP

The ARAP-Security AVP (AVP Code 73) is of type Unsigned32 and MAY be present in the AA-Answer message if the Framed-Protocol AVP (Section 4.4.10.1) is set to the value of ARAP, and the Result-Code AVP ([RFC6733], Section 7.1) is set to DIAMETER\_MULTI\_ROUND\_AUTH. See RFC 2869 for more information on the contents of this AVP.

#### 4.3.12. ARAP-Security-Data AVP

The ARAP-Security-Data AVP (AVP Code 74) is of type OctetString and MAY be present in the AA-Request or AA-Answer message if the Framed-Protocol AVP (Section 4.4.10.1) is set to the value of ARAP and the Result-Code AVP ([RFC6733], Section 7.1) is set to DIAMETER\_MULTI\_ROUND\_AUTH. This AVP contains the security module challenge or response associated with the ARAP Security Module specified in the ARAP-Security AVP (Section 4.3.11).

#### 4.4. NAS Authorization AVPs

This section contains the authorization AVPs supported in the NAS Application. The Service-Type AVP SHOULD be present in all messages and, based on its value, additional AVPs defined in this section and Section 4.5 MAY be present.

The following table gives the possible flag values for the session-level AVPs.

Attribute Name	Section Defined	AVP Flag rules	
		MUST	MUST NOT
Service-Type	4.4.1	M	V
Callback-Number	4.4.2	M	V
Callback-Id	4.4.3	M	V
Idle-Timeout	4.4.4	M	V
Port-Limit	4.4.5	M	V
NAS-Filter-Rule	4.4.6	M	V
Filter-Id	4.4.7	M	V
Configuration-Token	4.4.8	M	V
QoS-Filter-Rule	4.4.9		
Framed-Protocol	4.4.10.1	M	V
Framed-Routing	4.4.10.2	M	V
Framed-MTU	4.4.10.3	M	V
Framed-Compression	4.4.10.4	M	V
Framed-IP-Address	4.4.10.5.1	M	V
Framed-IP-Netmask	4.4.10.5.2	M	V
Framed-Route	4.4.10.5.3	M	V
Framed-Pool	4.4.10.5.4	M	V
Framed-Interface-Id	4.4.10.5.5	M	V
Framed-IPv6-Prefix	4.4.10.5.6	M	V
Framed-IPv6-Route	4.4.10.5.7	M	V
Framed-IPv6-Pool	4.4.10.5.8	M	V
Framed-IPX-Network	4.4.10.6.1	M	V
Framed-Appletalk-Link	4.4.10.7.1	M	V
Framed-Appletalk-Network	4.4.10.7.2	M	V
Framed-Appletalk-Zone	4.4.10.7.3	M	V
ARAP-Features	4.4.10.8.1	M	V
ARAP-Zone-Access	4.4.10.8.2	M	V
Login-IP-Host	4.4.11.1	M	V
Login-IPv6-Host	4.4.11.2	M	V
Login-Service	4.4.11.3	M	V
Login-TCP-Port	4.4.11.4.1	M	V
Login-LAT-Service	4.4.11.5.1	M	V
Login-LAT-Node	4.4.11.5.2	M	V
Login-LAT-Group	4.4.11.5.3	M	V
Login-LAT-Port	4.4.11.5.4	M	V



#### 4.4.1.1. Service-Type AVP

The Service-Type AVP (AVP Code 6) is of type Enumerated and contains the type of service the user has requested or the type of service to be provided. One such AVP MAY be present in an authentication and/or authorization request or response. A NAS is not required to implement all of these service types. It MUST treat unknown or unsupported Service-Types received in a response as a failure and end the session with a DIAMETER\_INVALID\_AVP\_VALUE Result-Code.

When used in a request, the Service-Type AVP SHOULD be considered a hint to the server that the NAS believes the user would prefer the kind of service indicated. The server is not required to honor the hint. Furthermore, if the service specified by the server is supported, but not compatible with the current mode of access, the NAS MUST fail to start the session. The NAS MUST also generate the appropriate error message(s).

The complete list of defined values that the Service-Type AVP can take can be found in Rigney, et al. [RFC2865] and the relevant IANA registry [RADIUSAttrVals], but the following values require further qualification here:

##### Login (1)

The user should be connected to a host. The message MAY include additional AVPs as defined in Section 4.4.11.4 or Section 4.4.11.5.

##### Framed (2)

A Framed Protocol, such as PPP or SLIP, should be started for the User. The message MAY include additional AVPs defined in Section 4.4.10, or Section 4.5 for tunneling services.

##### Callback Login (3)

The user should be disconnected and called back, then connected to a host. The message MAY include additional AVPs defined in this Section.

#### Callback Framed (4)

The user should be disconnected and called back, and then a Framed Protocol, such as PPP or SLIP, should be started for the user. The message MAY include additional AVPs defined in Section 4.4.10, or Section 4.5 for tunneling services.

#### 4.4.2. Callback-Number AVP

The Callback-Number AVP (AVP Code 19) is of type UTF8String and contains a dialing string to be used for callback, the format of which is deployment-specific. The Callback-Number AVP MAY be used in an authentication and/or authorization request as a hint to the server that a callback service is desired, but the server is not required to honor the hint in the corresponding response.

Any further codification of this field's allowed usage range is outside the scope of this specification.

#### 4.4.3. Callback-Id AVP

The Callback-Id AVP (AVP Code 20) is of type UTF8String and contains the name of a place to be called, to be interpreted by the NAS. This AVP MAY be present in an authentication and/or authorization response.

This AVP is not roaming-friendly as it assumes that the Callback-Id is configured on the NAS. Using the Callback-Number AVP (Section 4.4.2) is therefore RECOMMENDED.

#### 4.4.4. Idle-Timeout AVP

The Idle-Timeout AVP (AVP Code 28) is of type Unsigned32 and sets the maximum number of consecutive seconds of idle connection allowable to the user before termination of the session or before a prompt is issued. The default is none, or system specific.

#### 4.4.5. Port-Limit AVP

The Port-Limit AVP (AVP Code 62) is of type Unsigned32 and sets the maximum number of ports the NAS provides to the user. It MAY be used in an authentication and/or authorization request as a hint to the server that multilink PPP [RFC1990] service is desired, but the server is not required to honor the hint in the corresponding response.

#### 4.4.6. NAS-Filter-Rule AVP

The NAS-Filter-Rule AVP (AVP Code 400) is of type IPFilterRule and provides filter rules that need to be configured on the NAS for the user. One or more of these AVPs MAY be present in an authorization response.

#### 4.4.7. Filter-Id AVP

The Filter-Id AVP (AVP Code 11) is of type UTF8String and contains the name of the filter list for this user. It is intended to be human-readable. Zero or more Filter-Id AVPs MAY be sent in an authorization answer message.

Identifying a filter list by name allows the filter to be used on different NASes without regard to filter-list implementation details. However, this AVP is not roaming-friendly, as filter naming differs from one service provider to another.

In environments where backward compatibility with RADIUS is not required, it is RECOMMENDED that the NAS-Filter-Rule AVP (Section 4.4.6) be used instead.

#### 4.4.8. Configuration-Token AVP

The Configuration-Token AVP (AVP Code 78) is of type OctetString and is sent by a Diameter Server to a Diameter Proxy Agent in an AA-Answer command to indicate a type of user profile to be used. It should not be sent to a Diameter Client (NAS).

The format of the Data field of this AVP is site specific.

#### 4.4.9. QoS-Filter-Rule AVP

The QoS-Filter-Rule AVP (AVP Code 407) is of type QoSFilterRule (Section 4.1.1) and provides QoS filter rules that need to be configured on the NAS for the user. One or more such AVPs MAY be present in an authorization response.

The use of this AVP is NOT RECOMMENDED; the AVPs defined by Korhonen, et al. [RFC5777] SHOULD be used instead.

The following options are defined for the QoSFilterRule filters:

DSCP <color>

If action is set to tag (Section 4.1.1) this option MUST be included in the rule.

Color values are defined in Nichols, et al. [RFC2474]. Exact matching of DSCP values is required (no masks or ranges).

metering <rate> <color\_under> <color\_over>

The metering option provides Assured Forwarding, as defined in Heinanen, et al. [RFC2597]. and MUST be present if the action is set to meter (Section 4.1.1) The rate option is the throughput, in bits per second, used by the access device to mark packets. Traffic over the rate is marked with the color\_over codepoint, and traffic under the rate is marked with the color\_under codepoint. The color\_under and color\_over options contain the drop preferences and MUST conform to the recommended codepoint keywords described in RFC 2597 (e.g., AF13).

The metering option also supports the strict limit on traffic required by Expedited Forwarding, as defined in Davie, et al. [RFC3246]. The color\_over option may contain the keyword "drop" to prevent forwarding of traffic that exceeds the rate parameter.

#### 4.4.10. Framed Access Authorization AVPs

This section lists the authorization AVPs necessary to support framed access, such as PPP and SLIP. AVPs defined in this section MAY be present in a message if the Service-Type AVP was set to "Framed" or "Callback Framed".

##### 4.4.10.1. Framed-Protocol AVP

The Framed-Protocol AVP (AVP Code 7) is of type Enumerated and contains the framing to be used for framed access. This AVP MAY be present in both requests and responses. The supported values are listed in [RADIUSAttrVals].

##### 4.4.10.2. Framed-Routing AVP

The Framed-Routing AVP (AVP Code 10) is of type Enumerated and contains the routing method for the user when the user is a router to a network. This AVP SHOULD only be present in authorization responses. The supported values are listed in [RADIUSAttrVals].

#### 4.4.10.3. Framed-MTU AVP

The Framed-MTU AVP (AVP Code 12) is of type Unsigned32 and contains the Maximum Transmission Unit (MTU) to be configured for the user, when it is not negotiated by some other means (such as PPP). This AVP SHOULD only be present in authorization responses. The MTU value MUST be in the range from 64 to 65535.

#### 4.4.10.4. Framed-Compression AVP

The Framed-Compression AVP (AVP Code 13) is of type Enumerated and contains the compression protocol to be used for the link. It MAY be used in an authorization request as a hint to the server that a specific compression type is desired, but the server is not required to honor the hint in the corresponding response.

More than one compression protocol AVP MAY be sent. The NAS is responsible for applying the proper compression protocol to the appropriate link traffic.

The supported values are listed in [RADIUSAttrVals].

#### 4.4.10.5. IP Access Authorization AVPs

The AVPs defined in this section are used when the user requests, or is being granted, access service to IP.

##### 4.4.10.5.1. Framed-IP-Address AVP

The Framed-IP-Address AVP (AVP Code 8) [RFC2865] is of type OctetString and contains an IPv4 address of the type specified in the attribute value to be configured for the user. It MAY be used in an authorization request as a hint to the server that a specific address is desired, but the server is not required to honor the hint in the corresponding response.

Two values have special significance: 0xFFFFFFFF and 0xFFFFFFFFFE. The value 0xFFFFFFFF indicates that the NAS should allow the user to select an address (i.e., negotiated). The value 0xFFFFFFFFFE indicates that the NAS should select an address for the user (e.g., assigned from a pool of addresses kept by the NAS).

##### 4.4.10.5.2. Framed-IP-Netmask AVP

The Framed-IP-Netmask AVP (AVP Code 9) is of type OctetString and contains the four octets of the IPv4 netmask to be configured for the user when the user is a router to a network. It MAY be used in an authorization request as a hint to the server that a specific netmask

is desired, but the server is not required to honor the hint in the corresponding response. This AVP MUST be present in a response if the request included this AVP with a value of 0xFFFFFFFF.

#### 4.4.10.5.3. Framed-Route AVP

The Framed-Route AVP (AVP Code 22) is of type UTF8String and contains the 7-bit ASCII routing information to be configured for the user on the NAS. Zero or more of these AVPs MAY be present in an authorization response.

The string MUST contain a destination prefix in dotted quad form optionally followed by a slash and a decimal length specifier stating how many high-order bits of the prefix should be used. This is followed by a space, a gateway address in dotted quad form, a space, and one or more metrics separated by spaces; for example,

```
"192.0.2.0/24 192.0.2.1 1"
```

The length specifier may be omitted, in which case it should default to 8 bits for class A prefixes, to 16 bits for class B prefixes, and to 24 bits for class C prefixes; for example,

```
"192.0.2.0 192.0.2.1 1"
```

Whenever the gateway address is specified as "0.0.0.0" the IP address of the user SHOULD be used as the gateway address.

#### 4.4.10.5.4. Framed-Pool AVP

The Framed-Pool AVP (AVP Code 88) is of type OctetString and contains the name of an assigned address pool that SHOULD be used to assign an address for the user. If a NAS does not support multiple address pools, the NAS SHOULD ignore this AVP. Address pools are usually used for IP addresses but can be used for other protocols if the NAS supports pools for those protocols.

Although specified as type OctetString for compatibility with RADIUS [RFC2869], the encoding of the Data field SHOULD also conform to the rules for the UTF8String Data Format.

#### 4.4.10.5.5. Framed-Interface-Id AVP

The Framed-Interface-Id AVP (AVP Code 96) is of type Unsigned64 and contains the IPv6 interface identifier to be configured for the user. It MAY be used in authorization requests as a hint to the server that a specific interface id is desired, but the server is not required to honor the hint in the corresponding response.

#### 4.4.10.5.6. Framed-IPv6-Prefix AVP

The Framed-IPv6-Prefix AVP (AVP Code 97) is of type OctetString and contains the IPv6 prefix to be configured for the user. One or more AVPs MAY be used in authorization requests as a hint to the server that specific IPv6 prefixes are desired, but the server is not required to honor the hint in the corresponding response.

#### 4.4.10.5.7. Framed-IPv6-Route AVP

The Framed-IPv6-Route AVP (AVP Code 99) is of type UTF8String and contains the ASCII routing information to be configured for the user on the NAS. Zero or more of these AVPs MAY be present in an authorization response.

The string MUST contain an IPv6 address prefix followed by a slash and a decimal length specifier stating how many high order bits of the prefix should be used. This is followed by a space, a gateway address in hexadecimal notation, a space, and one or more metrics separated by spaces; for example,

```
"2001:db8::/32 2001:db8:106:a00:20ff:fe99:a998 1"
```

Whenever the gateway address is the IPv6 unspecified address, the IP address of the user SHOULD be used as the gateway address, such as in:

```
"2001:db8::/32 :: 1"
```

#### 4.4.10.5.8. Framed-IPv6-Pool AVP

The Framed-IPv6-Pool AVP (AVP Code 100) is of type OctetString and contains the name of an assigned pool that SHOULD be used to assign an IPv6 prefix for the user. If the access device does not support multiple prefix pools, it MUST ignore this AVP.

Although specified as type OctetString for compatibility with RADIUS [RFC3162], the encoding of the Data field SHOULD also conform to the rules for the UTF8String Data Format.

#### 4.4.10.6. IPX Access AVPs

The AVPs defined in this section are used when the user requests, or is being granted, access to an IPX network service [IPX].

##### 4.4.10.6.1. Framed-IPX-Network AVP

The Framed-IPX-Network AVP (AVP Code 23) is of type Unsigned32 and contains the IPX Network number to be configured for the user. It MAY be used in an authorization request as a hint to the server that a specific address is desired, but the server is not required to honor the hint in the corresponding response.

Two addresses have special significance: 0xFFFFFFFF and 0xFFFFFFFFE. The value 0xFFFFFFFF indicates that the NAS should allow the user to select an address (i.e., Negotiated). The value 0xFFFFFFFFE indicates that the NAS should select an address for the user (e.g., assign it from a pool of one or more IPX networks kept by the NAS).

#### 4.4.10.7. AppleTalk Network Access AVPs

The AVPs defined in this section are used when the user requests, or is being granted, access to an AppleTalk network [AppleTalk].

##### 4.4.10.7.1. Framed-AppleTalk-Link AVP

The Framed-AppleTalk-Link AVP (AVP Code 37) is of type Unsigned32 and contains the AppleTalk network number that should be used for the serial link to the user, which is another AppleTalk router. This AVP MUST only be present in an authorization response and is never used when the user is not another router.

Despite the size of the field, values range from 0 to 65,535. The special value of 0 indicates an unnumbered serial link. A value of 1 to 65,535 means that the serial line between the NAS and the user should be assigned that value as an AppleTalk network number.

##### 4.4.10.7.2. Framed-AppleTalk-Network AVP

The Framed-AppleTalk-Network AVP (AVP Code 38) is of type Unsigned32 and contains the AppleTalk Network number that the NAS should probe to allocate an AppleTalk node for the user. This AVP MUST only be present in an authorization response and is never used when the user is not another router. Multiple instances of this AVP indicate that the NAS may probe, using any of the network numbers specified.

Despite the size of the field, values range from 0 to 65,535. The special value 0 indicates that the NAS should assign a network for the user, using its default cable range. A value between 1 and 65,535 (inclusive) indicates to the AppleTalk Network that the NAS should probe to find an address for the user.

##### 4.4.10.7.3. Framed-AppleTalk-Zone AVP



The Framed-AppleTalk-Zone AVP (AVP Code 39) is of type OctetString and contains the AppleTalk Default Zone to be used for this user. This AVP MUST only be present in an authorization response. Multiple instances of this AVP in the same message are not allowed.

The codification of this field's allowed range is outside the scope of this specification.

#### 4.4.10.8. AppleTalk Remote Access AVPs

The AVPs defined in this section are used when the user requests, or is being granted, access to the AppleTalk network via the AppleTalk Remote Access Protocol [ARAP]. They are only present if the Framed-Protocol AVP (Section 4.4.10.1) is set to ARAP. Section 2.2 of RFC 2869 describes the operational use of these attributes.

##### 4.4.10.8.1. ARAP-Features AVP

The ARAP-Features AVP (AVP Code 71) is of type OctetString and MAY be present in the AA-Accept message if the Framed-Protocol AVP is set to the value of ARAP. See RFC 2869 for more information about the format of this AVP.

##### 4.4.10.8.2. ARAP-Zone-Access AVP

The ARAP-Zone-Access AVP (AVP Code 72) is of type Enumerated and MAY be present in the AA-Accept message if the Framed-Protocol AVP is set to the value of ARAP.

The supported values are listed in [RADIUSAttrVals] and defined in RFC 2869.

#### 4.4.11. Non-Framed Access Authorization AVPs

This section contains the authorization AVPs that are needed to support terminal server functionality. AVPs defined in this section MAY be present in a message if the Service-Type AVP was set to "Login" or "Callback Login".

##### 4.4.11.1. Login-IP-Host AVP

The Login-IP-Host AVP (AVP Code 14) [RFC2865] is of type OctetString and contains the IPv4 address of a host with which to connect the user when the Login-Service AVP is included. It MAY be used in an AA-Request command as a hint to the Diameter Server that a specific host is desired, but the Diameter Server is not required to honor the hint in the AA-Answer.

Two addresses have special significance: all ones and 0. The value of all ones indicates that the NAS SHOULD allow the user to select an address. The value 0 indicates that the NAS SHOULD select a host to connect the user to.

#### 4.4.11.2. Login-IPv6-Host AVP

The Login-IPv6-Host AVP (AVP Code 98) [RFC3162] is of type OctetString and contains the IPv6 address of a host with which to connect the user when the Login-Service AVP is included. It MAY be used in an AA-Request command as a hint to the Diameter Server that a specific host is desired, but the Diameter Server is not required to honor the hint in the AA-Answer.

Two addresses have special significance, 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF and 0. The value 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF indicates that the NAS SHOULD allow the user to select an address. The value 0 indicates that the NAS SHOULD select a host to connect the user to.

#### 4.4.11.3. Login-Service AVP

The Login-Service AVP (AVP Code 15) is of type Enumerated and contains the service that should be used to connect the user to the login host. This AVP SHOULD only be present in authorization responses. The supported values are listed in RFC 2869.

#### 4.4.11.4. TCP Services

The AVP described in the following section MAY be present if the Login-Service AVP is set to Telnet, Rlogin, TCP Clear, or TCP Clear Quiet.

##### 4.4.11.4.1. Login-TCP-Port AVP

The Login-TCP-Port AVP (AVP Code 16) is of type Unsigned32 and contains the TCP port with which the user is to be connected when the Login-Service AVP is also present. This AVP SHOULD only be present in authorization responses. The value MUST NOT be greater than 65,535.

#### 4.4.11.5. LAT Services

The AVPs described in this section MAY be present if the Login-Service AVP is set to LAT [LAT].

##### 4.4.11.5.1. Login-LAT-Service AVP

The Login-LAT-Service AVP (AVP Code 34) is of type OctetString and contains the system with which the user is to be connected by LAT. It MAY be used in an authorization request as a hint to the server that a specific service is desired, but the server is not required to honor the hint in the corresponding response. This AVP MUST only be present in the response if the Login-Service AVP states that LAT is desired.

Administrators use this service attribute when dealing with clustered systems. In these environments, several different time-sharing hosts share the same resources (disks, printers, etc.), and administrators often configure each host to offer access (service) to each of the shared resources. In this case, each host in the cluster advertises its services through LAT broadcasts.

Sophisticated users often know which service providers (machines) are faster and tend to use a node name when initiating a LAT connection. Some administrators want particular users to use certain machines as a primitive form of load balancing (although LAT knows how to do load balancing itself).

The String field contains the identity of the LAT service to use. The LAT Architecture allows this string to contain \$ (dollar), - (hyphen), . (period), \_ (underscore), numerics, upper- and lowercase alphabets, and the ISO Latin-1 character set extension [ISO.8859-1.1987]. All LAT string comparisons are case insensitive.

#### 4.4.11.5.2. Login-LAT-Node AVP

The Login-LAT-Node AVP (AVP Code 35) is of type OctetString and contains the Node with which the user is to be automatically connected by LAT. It MAY be used in an authorization request as a hint to the server that a specific LAT node is desired, but the server is not required to honor the hint in the corresponding response. This AVP MUST only be present in a response if the Login-Service-Type AVP is set to LAT.

The String field contains the identity of the LAT service to use. The LAT Architecture allows this string to contain \$ (dollar), - (hyphen), . (period), \_ (underscore), numerics, upper- and lowercase alphabets, and the ISO Latin-1 character set extension [ISO.8859-1.1987]. All LAT string comparisons are case insensitive.

#### 4.4.11.5.3. Login-LAT-Group AVP

The Login-LAT-Group AVP (AVP Code 36) is of type OctetString and contains a string identifying the LAT group codes this user is authorized to use. It MAY be used in an authorization request as a hint to the server that a specific group is desired, but the server is not required to honor the hint in the corresponding response. This AVP MUST only be present in a response if the Login-Service-Type AVP is set to LAT.

LAT supports 256 different group codes, which LAT uses as a form of access rights. LAT encodes the group codes as a 256-bit bitmap.

Administrators can assign one or more of the group code bits at the LAT service provider; it will only accept LAT connections that have these group codes set in the bitmap. The administrators assign a bitmap of authorized group codes to each user. LAT gets these from the operating system and uses them in its requests to the service providers.

The codification of the range of allowed usage of this field is outside the scope of this specification.

#### 4.4.11.5.4. Login-LAT-Port AVP

The Login-LAT-Port AVP (AVP Code 63) is of type OctetString and contains the Port with which the user is to be connected by LAT. It MAY be used in an authorization request as a hint to the server that a specific port is desired, but the server is not required to honor the hint in the corresponding response. This AVP MUST only be present in a response if the Login-Service-Type AVP is set to LAT.

The String field contains the identity of the LAT service to use. The LAT Architecture allows this string to contain \$ (dollar), - (hyphen), . (period), \_ (underscore), numerics, upper- and lower-case alphabets, and the ISO Latin-1 character set extension [ISO.8859-1.1987].

All LAT string comparisons are case insensitive.

#### 4.5. NAS Tunneling AVPs

Some NASes support compulsory tunnel services in which the incoming connection data is conveyed by an encapsulation method to a gateway elsewhere in the network. This is typically transparent to the service user, and the tunnel characteristics may be described by the remote AAA server, based on the user's authorization information. Several tunnel characteristics may be returned, and the NAS

implementation may choose one. See Zorn, et al. [RFC2868] and Zorn, Aboba & Mitton [RFC2867] for further information.

The following table gives the possible flag values for the session level AVPs and specifies whether the AVP MAY be encrypted.

		+-----+	
		AVP Flag rules	
		-----+	
Attribute Name	Section Defined	MUST	MUST NOT
		-----	-----
Tunneling	4.5.1	M	V
Tunnel-Type	4.5.2	M	V
Tunnel-Medium-Type	4.5.3	M	V
Tunnel-Client-Endpoint	4.5.4	M	V
Tunnel-Server-Endpoint	4.5.5	M	V
Tunnel-Password	4.5.6	M	V
Tunnel-Private-Group-Id	4.5.7	M	V
Tunnel-Assignment-Id	4.5.8	M	V
Tunnel-Preference	4.5.9	M	V
Tunnel-Client-Auth-Id	4.5.10	M	V
Tunnel-Server-Auth-Id	4.5.11	M	V
-----		-----	-----

#### 4.5.1. Tunneling AVP

The Tunneling AVP (AVP Code 401) is of type Grouped and contains the following AVPs, used to describe a compulsory tunnel service ([RFC2868], [RFC2867]). Its data field has the following ABNF grammar:

```
Tunneling ::= < AVP Header: 401 >
              { Tunnel-Type }
              { Tunnel-Medium-Type }
              { Tunnel-Client-Endpoint }
              { Tunnel-Server-Endpoint }
              [ Tunnel-Preference ]
              [ Tunnel-Client-Auth-Id ]
              [ Tunnel-Server-Auth-Id ]
              [ Tunnel-Assignment-Id ]
              [ Tunnel-Password ]
              [ Tunnel-Private-Group-Id ]
```

#### 4.5.2. Tunnel-Type AVP

The Tunnel-Type AVP (AVP Code 64) is of type Enumerated and contains the tunneling protocol(s) to be used (in the case of a tunnel initiator) or in use (in the case of a tunnel terminator). It MAY be used in an authorization request as a hint to the server that a specific tunnel type is desired, but the server is not required to honor the hint in the corresponding response.

The Tunnel-Type AVP SHOULD also be included in ACR messages.

A tunnel initiator is not required to implement any of these tunnel types. If a tunnel initiator receives a response that contains only unknown or unsupported Tunnel-Types, the tunnel initiator MUST behave as though a response were received with the Result-Code indicating a failure.

The supported values are listed in [RADIUSAttrVals].

#### 4.5.3. Tunnel-Medium-Type AVP

The Tunnel-Medium-Type AVP (AVP Code 65) is of type Enumerated and contains the transport medium to use when creating a tunnel for protocols (such as L2TP [RFC3931]) that can operate over multiple transports. It MAY be used in an authorization request as a hint to the server that a specific medium is desired, but the server is not required to honor the hint in the corresponding response.

The supported values are listed in [RADIUSAttrVals].

#### 4.5.4. Tunnel-Client-Endpoint AVP

The Tunnel-Client-Endpoint AVP (AVP Code 66) is of type UTF8String and contains the address of the initiator end of the tunnel. It MAY be used in an authorization request as a hint to the server that a specific endpoint is desired, but the server is not required to honor the hint in the corresponding response. This AVP SHOULD be included in the corresponding ACR messages, in which case it indicates the address from which the tunnel was initiated. This AVP, along with the Tunnel-Server-Endpoint (Section 4.5.5) and Session-Id AVPs ([RFC6733], Section 8.8), can be used to provide a globally unique means to identify a tunnel for accounting and auditing purposes.

If the value of the Tunnel-Medium-Type AVP (Section 4.5.3) is IPv4 (1), then this string is either the fully qualified domain name (FQDN) of the tunnel client machine, or a "dotted-decimal" IP address. Implementations MUST support the dotted-decimal format and SHOULD support the FQDN format for IP addresses.

If Tunnel-Medium-Type is IPv6 (2), then this string is either the FQDN of the tunnel client machine, or a text representation of the address in either the preferred or alternate form [RFC3516]. Conforming implementations MUST support the preferred form and SHOULD support both the alternate text form and the FQDN format for IPv6 addresses.

If Tunnel-Medium-Type is neither IPv4 nor IPv6, then this string is a tag referring to configuration data local to the Diameter client that describes the interface or medium-specific client address to use.

Note that this application handles internationalized domain names in the same way as the Diameter base protocol (see Appendix D of RFC 6733 for details).

#### 4.5.5. Tunnel-Server-Endpoint AVP

The Tunnel-Server-Endpoint AVP (AVP Code 67) is of type UTF8String and contains the address of the server end of the tunnel. It MAY be used in an authorization request as a hint to the server that a specific endpoint is desired, but the server is not required to honor the hint in the corresponding response.

This AVP SHOULD be included in the corresponding ACR messages, in which case it indicates the address from which the tunnel was initiated. This AVP, along with the Tunnel-Client-Endpoint (Section 4.5.4) and Session-Id AVP ([RFC6733], Section 8.8), can be used to provide a globally unique means to identify a tunnel for accounting and auditing purposes.

If Tunnel-Medium-Type is IPv4 (1), then this string is either the fully qualified domain name (FQDN) of the tunnel server machine, or a "dotted-decimal" IP address. Implementations MUST support the dotted-decimal format and SHOULD support the FQDN format for IP addresses.

If Tunnel-Medium-Type is IPv6 (2), then this string is either the FQDN of the tunnel server machine, or a text representation of the address in either the preferred or alternate form [RFC3516]. Implementations MUST support the preferred form and SHOULD support both the alternate text form and the FQDN format for IPv6 addresses.

If Tunnel-Medium-Type is not IPv4 or IPv6, this string is a tag referring to configuration data local to the Diameter client that describes the interface or medium-specific server address to use.

Note that this application handles internationalized domain names in the same way as the Diameter base protocol (see Appendix D of RFC 6733 for details).

#### 4.5.6. Tunnel-Password AVP

The Tunnel-Password AVP (AVP Code 69) is of type OctetString and may contain a password to be used to authenticate to a remote server.

The Tunnel-Password AVP SHOULD NOT be used in untrusted proxy environments without encrypting it by using end-to-end security techniques.

#### 4.5.7. Tunnel-Private-Group-Id AVP

The Tunnel-Private-Group-Id AVP (AVP Code 81) is of type OctetString and contains the group Id for a particular tunneled session. The Tunnel-Private-Group-Id AVP MAY be included in an authorization request if the tunnel initiator can predetermine the group resulting from a particular connection. It SHOULD be included in the authorization response if this tunnel session is to be treated as belonging to a particular private group. Private groups may be used to associate a tunneled session with a particular group of users. For example, it MAY be used to facilitate routing of unregistered IP addresses through a particular interface. This AVP SHOULD be included in the ACR messages that pertain to the tunneled session.

#### 4.5.8. Tunnel-Assignment-Id AVP

The Tunnel-Assignment-Id AVP (AVP Code 82) is of type OctetString and is used to indicate to the tunnel initiator the particular tunnel to which a session is to be assigned. Some tunneling protocols, such as PPTP [RFC2637] and L2TP [RFC3931], allow for sessions between the same two tunnel endpoints to be multiplexed over the same tunnel and also for a given session to use its own dedicated tunnel. This attribute provides a mechanism for Diameter to inform the tunnel initiator (for example, a LAC) whether to assign the session to a multiplexed tunnel or to a separate tunnel. Furthermore, it allows for sessions sharing multiplexed tunnels to be assigned to different multiplexed tunnels.

A particular tunneling implementation may assign differing characteristics to particular tunnels. For example, different tunnels may be assigned different QoS parameters. Such tunnels may be used to carry either individual or multiple sessions. The Tunnel-Assignment-Id attribute thus allows the Diameter server to indicate that a particular session is to be assigned to a tunnel providing an appropriate level of service. It is expected that any QoS-related



Diameter tunneling attributes defined in the future accompanying this one will be associated by the tunnel initiator with the Id given by this attribute. In the meantime, any semantic given to a particular Id string is a matter left to local configuration in the tunnel initiator.

The Tunnel-Assignment-Id AVP is of significance only to Diameter and the tunnel initiator. The Id it specifies is only intended to be of local use to Diameter and the tunnel initiator. The Id assigned by the tunnel initiator is not conveyed to the tunnel peer.

This attribute MAY be included in authorization responses. The tunnel initiator receiving this attribute MAY choose to ignore it and to assign the session to an arbitrary multiplexed or non-multiplexed tunnel between the desired endpoints. This AVP SHOULD also be included in the Accounting-Request messages pertaining to the tunneled session.

If a tunnel initiator supports the Tunnel-Assignment-Id AVP, then it should assign a session to a tunnel in the following manner:

- o If this AVP is present and a tunnel exists between the specified endpoints with the specified Id, then the session should be assigned to that tunnel.
- o If this AVP is present and no tunnel exists between the specified endpoints with the specified Id, then a new tunnel should be established for the session and the specified Id should be associated with the new tunnel.
- o If this AVP is not present, then the session is assigned to an unnamed tunnel. If an unnamed tunnel does not yet exist between the specified endpoints, then it is established and used for this session and for subsequent ones established without the Tunnel-Assignment-Id attribute. A tunnel initiator MUST NOT assign a session for which a Tunnel-Assignment-Id AVP was not specified to a named tunnel (i.e., one that was initiated by a session specifying this AVP).

Note that the same Id may be used to name different tunnels if these tunnels are between different endpoints.

#### 4.5.9. Tunnel-Preference AVP

The Tunnel-Preference AVP (AVP Code 83) is of type Unsigned32 and is used to identify the relative preference assigned to each tunnel when more than one set of tunneling AVPs is returned within separate Grouped-AVP AVPs. It MAY be used in an authorization request as a

hint to the server that a specific preference is desired, but the server is not required to honor the hint in the corresponding response.

For example, suppose that AVPs describing two tunnels are returned by the server, one with a Tunnel-Type of PPTP and the other with a Tunnel-Type of L2TP. If the tunnel initiator supports only one of the Tunnel-Types returned, it will initiate a tunnel of that type. If, however, it supports both tunnel protocols, it SHOULD use the value of the Tunnel-Preference AVP to decide which tunnel should be started. The tunnel with the lowest numerical value in the Value field of this AVP SHOULD be given the highest preference. The values assigned to two or more instances of the Tunnel-Preference AVP within a given authorization response MAY be identical. In this case, the tunnel initiator SHOULD use locally configured metrics to decide which set of AVPs to use.

#### 4.5.10. Tunnel-Client-Auth-Id AVP

The Tunnel-Client-Auth-Id AVP (AVP Code 90) is of type UTF8String and specifies the 7-bit US-ASCII name used by the tunnel initiator during the authentication phase of tunnel establishment. It MAY be used in an authorization request as a hint to the server that a specific preference is desired, but the server is not required to honor the hint in the corresponding response. This AVP MUST be present in the authorization response if an authentication name other than the default is desired. This AVP SHOULD be included in the ACR messages pertaining to the tunneled session.

#### 4.5.11. Tunnel-Server-Auth-Id AVP

The Tunnel-Server-Auth-Id AVP (AVP Code 91) is of type UTF8String and specifies the 7-bit US-ASCII name used by the tunnel terminator during the authentication phase of tunnel establishment. It MAY be used in an authorization request as a hint to the server that a specific preference is desired, but the server is not required to honor the hint in the corresponding response. This AVP MUST be present in the authorization response if an authentication name other than the default is desired. This AVP SHOULD be included in the ACR messages pertaining to the tunneled session.

#### 4.6. NAS Accounting AVPs

Applications implementing this specification use Diameter Accounting (as defined in [RFC6733]) and the AVPs in the following section. Service-specific AVP usage is defined in the tables in Section 5.

If accounting is active, Accounting Request (ACR) messages SHOULD be sent after the completion of any Authentication or Authorization transaction and at the end of a Session. The value of the Accounting-Record-Type AVP [RFC6733] indicates the type of event. All other AVPs identify the session and provide additional information relevant to the event.

The successful completion of the first Authentication or Authorization transaction SHOULD cause a START\_RECORD to be sent. If additional Authentications or Authorizations occur in later transactions, the first exchange should generate a START\_RECORD, and the later an INTERIM\_RECORD. For a given session, there MUST only be one set of matching START and STOP records, with any number of INTERIM\_RECORDS in between, or one EVENT\_RECORD indicating the reason a session wasn't started.

The following table gives the possible flag values for the session level AVPs and specifies whether the AVP MAY be encrypted.

Attribute Name	Section Defined	AVP Flag rules	
		MUST	MUST NOT
Accounting-Input-Octets	4.6.1	M	V
Accounting-Output-Octets	4.6.2	M	V
Accounting-Input-Packets	4.6.3	M	V
Accounting-Output-Packets	4.6.4	M	V
Acct-Session-Time	4.6.5	M	V
Acct-Authentic	4.6.6	M	V
Accounting-Auth-Method	4.6.7	M	V
Acct-Delay-Time	4.6.8	M	V
Acct-Link-Count	4.6.9	M	V
Acct-Tunnel-Connection	4.6.10	M	V
Acct-Tunnel-Packets-Lost	4.6.11	M	V

#### 4.6.1. Accounting-Input-Octets AVP

The Accounting-Input-Octets AVP (AVP Code 363) is of type Unsigned64 and contains the number of octets received from the user.

For NAS usage, this AVP indicates how many octets have been received from the port in the course of this session. It can only be present in ACR messages with an Accounting-Record-Type [RFC6733] of INTERIM\_RECORD or STOP\_RECORD.

#### 4.6.2. Accounting-Output-Octets AVP

The Accounting-Output-Octets AVP (AVP Code 364) is of type Unsigned64 and contains the number of octets sent to the user.

For NAS usage, this AVP indicates how many octets have been sent to the port in the course of this session. It can only be present in ACR messages with an Accounting-Record-Type of INTERIM\_RECORD or STOP\_RECORD.

#### 4.6.3. Accounting-Input-Packets AVP

The Accounting-Input-Packets (AVP Code 365) is of type Unsigned64 and contains the number of packets received from the user.

For NAS usage, this AVP indicates how many packets have been received from the port over the course of a session being provided to a Framed User. It can only be present in ACR messages with an Accounting-Record-Type of INTERIM\_RECORD or STOP\_RECORD.

#### 4.6.4. Accounting-Output-Packets AVP

The Accounting-Output-Packets (AVP Code 366) is of type Unsigned64 and contains the number of IP packets sent to the user.

For NAS usage, this AVP indicates how many packets have been sent to the port over the course of a session being provided to a Framed User. It can only be present in ACR messages with an Accounting-Record-Type of INTERIM\_RECORD or STOP\_RECORD.

#### 4.6.5. Acct-Session-Time AVP

The Acct-Session-Time AVP (AVP Code 46) is of type Unsigned32 and indicates the length of the current session in seconds. It can only be present in ACR messages with an Accounting-Record-Type of INTERIM\_RECORD or STOP\_RECORD.

#### 4.6.6. Acct-Authentic AVP

The Acct-Authentic AVP (AVP Code 45) is of type Enumerated and specifies how the user was authenticated. The supported values are listed in [RADIUSAttrVals].

#### 4.6.7. Accounting-Auth-Method AVP

The Accounting-Auth-Method AVP (AVP Code 406) is of type Enumerated. A NAS MAY include this AVP in an Accounting-Request message to indicate the method used to authenticate the user. (Note that this AVP is semantically equivalent, and the supported values are identical, to the Microsoft MS-Acct-Auth-Type vendor-specific RADIUS attribute [RFC2548]).

#### 4.6.8. Acct-Delay-Time AVP

The Acct-Delay-Time AVP (AVP Code 41) is of type Unsigned32 and indicates the number of seconds the Diameter client has been trying to send the Accounting-Request (ACR). The accounting server may subtract this value from the time when the ACR arrives at the server to calculate the approximate time of the event that caused the ACR to be generated.

This AVP is not used for retransmissions at the transport level (TCP or SCTP). Rather, it may be used when an ACR command cannot be transmitted because there is no appropriate peer to transmit it to or was rejected because it could not be delivered. In these cases, the command MAY be buffered and transmitted later, when an appropriate peer-connection is available or after sufficient time has passed that the destination-host may be reachable and operational. If the ACR is re-sent in this way, the Acct-Delay-Time AVP SHOULD be included. The value of this AVP indicates the number of seconds that elapsed between the time of the first attempt at transmission and the current attempt.

#### 4.6.9. Acct-Link-Count AVP

The Acct-Link-Count AVP (AVP Code 51) is of type Unsigned32 and indicates the total number of links that have been active (current or closed) in a given multilink session at the time the accounting record is generated. This AVP MAY be included in Accounting-Requests for any session that may be part of a multilink service.

The Acct-Link-Count AVP may be used to make it easier for an accounting server to know when it has all the records for a given multilink service. When the number of Accounting-Requests received with Accounting-Record-Type = STOP\_RECORD and with the same Acct-Multi-Session-Id and unique Session-Ids equals the largest value of Acct-Link-Count seen in those Accounting-Requests, all STOP\_RECORD Accounting-Requests for that multilink service have been received.

The following example, showing eight Accounting-Requests, illustrates how the Acct-Link-Count AVP is used. In the table below, only the

relevant AVPs are shown, although additional AVPs containing accounting information will be present in the Accounting-Requests.

Acct-Multi- Session-Id	Session-Id	Accounting- Record-Type	Acct- Link-Count
"...10"	"...10"	START_RECORD	1
"...10"	"...11"	START_RECORD	2
"...10"	"...11"	STOP_RECORD	2
"...10"	"...12"	START_RECORD	3
"...10"	"...13"	START_RECORD	4
"...10"	"...12"	STOP_RECORD	4
"...10"	"...13"	STOP_RECORD	4
"...10"	"...10"	STOP_RECORD	4

#### 4.6.10. Acct-Tunnel-Connection AVP

The Acct-Tunnel-Connection AVP (AVP Code 68) is of type OctetString and contains the identifier assigned to the tunnel session. This AVP, along with the Tunnel-Client-Endpoint (Section 4.5.4) and Tunnel-Server-Endpoint (Section 4.5.5) AVPs, may be used to provide a means to uniquely identify a tunnel session for auditing purposes.

The format of the identifier in this AVP depends upon the value of the Tunnel-Type AVP (Section 4.5.2). For example, to identify an L2TP tunnel connection fully, the L2TP Tunnel Id and Call Id might be encoded in this field. The exact encoding of this field is implementation dependent.

#### 4.6.11. Acct-Tunnel-Packets-Lost AVP

The Acct-Tunnel-Packets-Lost AVP (AVP Code 86) is of type Unsigned32 and contains the number of packets lost on a given tunnel.

### 5. AVP Occurrence Tables

The following tables present the AVPs used by NAS applications in NAS messages and specify in which Diameter messages they may or may not be present. Messages and AVPs defined in the base Diameter protocol [RFC6733] are not described in this document. Note that AVPs that can only be present within a Grouped AVP are not represented in this table.

The tables use 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.
- 1 Exactly one instance of the AVP MUST be present in the message.

#### 5.1. AA-Request/Answer AVP Table

The table in this section is limited to the Command Codes defined in this specification.

AVP Name	Command	
	AAR	AAA
Acct-Interim-Interval	0	0-1
ARAP-Challenge-Response	0	0-1
ARAP-Features	0	0-1
ARAP-Password	0-1	0
ARAP-Security	0-1	0-1
ARAP-Security-Data	0+	0+
ARAP-Zone-Access	0	0-1
Auth-Application-Id	1	1
Auth-Grace-Period	0-1	0-1
Auth-Request-Type	1	1
Auth-Session-State	0-1	0-1
Authorization-Lifetime	0-1	0-1

Attribute Name	Command	
	AAR	AAA
Callback-Id	0	0-1
Callback-Number	0-1	0-1
Called-Station-Id	0-1	0
Calling-Station-Id	0-1	0
CHAP-Auth	0-1	0
CHAP-Challenge	0-1	0
Class	0	0+
Configuration-Token	0	0+
Connect-Info	0+	0
Destination-Host	0-1	0

Destination-Realm	1	0
Error-Message	0	0-1
Error-Reporting-Host	0	0-1
Failed-AVP	0+	0+
Filter-Id	0	0+
Framed-Appletalk-Link	0	0-1
Framed-Appletalk-Network	0	0+
Framed-Appletalk-Zone	0	0-1
Framed-Compression	0+	0+
Framed-Interface-Id	0-1	0-1
Framed-IP-Address	0-1	0-1
Framed-IP-Netmask	0-1	0-1
Framed-IPv6-Prefix	0+	0+
Framed-IPv6-Pool	0	0-1
Framed-IPv6-Route	0	0+
Framed-IPX-Network	0	0-1
Framed-MTU	0-1	0-1
Framed-Pool	0	0-1
Framed-Protocol	0-1	0-1
Framed-Route	0	0+
Framed-Routing	0	0-1
Idle-Timeout	0	0-1
Login-IP-Host	0+	0+
Login-IPv6-Host	0+	0+
Login-LAT-Group	0-1	0-1
Login-LAT-Node	0-1	0-1
Login-LAT-Port	0-1	0-1
Login-LAT-Service	0-1	0-1
Login-Service	0	0-1
Login-TCP-Port	0	0-1
Multi-Round-Time-Out	0	0-1
-----+-----		

Attribute Name	Command	
	AAR	AAA
-----+-----		
NAS-Filter-Rule	0	0+
NAS-Identifier	0-1	0
NAS-IP-Address	0-1	0
NAS-IPv6-Address	0-1	0
NAS-Port	0-1	0
NAS-Port-Id	0-1	0
NAS-Port-Type	0-1	0
Origin-AAA-Protocol	0-1	0-1
Origin-Host	1	1
Origin-Realm	1	1



Origin-State-Id	0-1	0-1
Originating-Line-Info	0-1	0
Password-Retry	0	0-1
Port-Limit	0-1	0-1
Prompt	0	0-1
Proxy-Info	0+	0+
QoS-Filter-Rule	0	0+
Re-Auth-Request-Type	0	0-1
Redirect-Host	0	0+
Redirect-Host-Usage	0	0-1
Redirect-Max-Cache-Time	0	0-1
Reply-Message	0	0+
Result-Code	0	1
Route-Record	0+	0
Service-Type	0-1	0-1
Session-Id	1	1
Session-Timeout	0	0-1
State	0-1	0-1
Tunneling	0+	0+
User-Name	0-1	0-1
User-Password	0-1	0
-----	-----	-----

## 5.2. Accounting AVP Tables

The tables in this section are used to show which AVPs defined in this document are to be present and used in NAS application Accounting messages. These AVPs are defined in this document, as well as in [RFC6733] and [RFC2866].

### 5.2.1. Framed Access Accounting AVP Table

The table in this section is used when the Service-Type AVP (Section 4.4.1) specifies Framed Access.

Attribute Name	Command	
	ACR	ACA
Accounting-Auth-Method	0-1	0
Accounting-Input-Octets	1	0
Accounting-Input-Packets	1	0
Accounting-Output-Octets	1	0
Accounting-Output-Packets	1	0
Accounting-Record-Number	0-1	0-1
Accounting-Record-Type	1	1
Accounting-Realtime-Required	0-1	0-1

Accounting-Sub-Session-Id	0-1	0-1
Acct-Application-Id	0-1	0-1
Acct-Session-Id	1	0-1
Acct-Multi-Session-Id	0-1	0-1
Acct-Authentic	1	0
Acct-Delay-Time	0-1	0
Acct-Interim-Interval	0-1	0-1
Acct-Link-Count	0-1	0
Acct-Session-Time	1	0
Acct-Tunnel-Connection	0-1	0
Acct-Tunnel-Packets-Lost	0-1	0
Authorization-Lifetime	0-1	0
Callback-Id	0-1	0
Callback-Number	0-1	0
Called-Station-Id	0-1	0
Calling-Station-Id	0-1	0
Class	0+	0+
Connection-Info	0+	0
Destination-Host	0-1	0
Destination-Realm	1	0
Event-Timestamp	0-1	0-1
Error-Message	0	0-1
Error-Reporting-Host	0	0-1
Failed-AVP	0	0+

Attribute Name	Command	
	ACR	ACA
Framed-AppleTalk-Link	0-1	0
Framed-AppleTalk-Network	0-1	0
Framed-AppleTalk-Zone	0-1	0
Framed-Compression	0-1	0
Framed-IP-Address	0-1	0
Framed-IP-Netmask	0-1	0
Framed-IPv6-Prefix	0+	0
Framed-IPv6-Pool	0-1	0
Framed-IPX-Network	0-1	0
Framed-MTU	0-1	0
Framed-Pool	0-1	0
Framed-Protocol	0-1	0
Framed-Route	0-1	0
Framed-Routing	0-1	0
NAS-Filter-Rule	0+	0
NAS-Identifier	0-1	0-1
NAS-IP-Address	0-1	0-1

NAS-IPv6-Address	0-1	0-1
NAS-Port	0-1	0-1
NAS-Port-Id	0-1	0-1
NAS-Port-Type	0-1	0-1
Origin-AAA-Protocol	0-1	0-1
Origin-Host	1	1
Origin-Realm	1	1
Origin-State-Id	0-1	0-1
Originating-Line-Info	0-1	0
Proxy-Info	0+	0+
QoS-Filter-Rule	0+	0
Route-Record	0+	0
Result-Code	0	1
Service-Type	0-1	0-1
Session-Id	1	1
Termination-Cause	0-1	0-1
Tunnel-Assignment-Id	0-1	0
Tunnel-Client-Endpoint	0-1	0
Tunnel-Medium-Type	0-1	0
Tunnel-Private-Group-Id	0-1	0
Tunnel-Server-Endpoint	0-1	0
Tunnel-Type	0-1	0
User-Name	0-1	0-1
-----+-----+		

#### 5.2.2. Non-Framed Access Accounting AVP Table

The table in this section is used when the Service-Type AVP (Section 4.4.1) specifies Non-Framed Access.

Attribute Name	Command	
	ACR	ACA
Accounting-Auth-Method	0-1	0
Accounting-Input-Octets	1	0
Accounting-Output-Octets	1	0
Accounting-Record-Type	1	1
Accounting-Record-Number	0-1	0-1
Accounting-Realtime-Required	0-1	0-1
Accounting-Sub-Session-Id	0-1	0-1
Acct-Application-Id	0-1	0-1
Acct-Session-Id	1	0-1
Acct-Multi-Session-Id	0-1	0-1
Acct-Authentic	1	0
Acct-Delay-Time	0-1	0
Acct-Interim-Interval	0-1	0-1

Acct-Link-Count	0-1	0
Acct-Session-Time	1	0
Authorization-Lifetime	0-1	0
Callback-Id	0-1	0
Callback-Number	0-1	0
Called-Station-Id	0-1	0
Calling-Station-Id	0-1	0
Class	0+	0+
Connection-Info	0+	0
Destination-Host	0-1	0
Destination-Realm	1	0
Event-Timestamp	0-1	0-1
Error-Message	0	0-1
Error-Reporting-Host	0	0-1
Failed-AVP	0	0+
Login-IP-Host	0+	0
Login-IPv6-Host	0+	0
Login-LAT-Service	0-1	0
Login-LAT-Node	0-1	0
Login-LAT-Group	0-1	0
Login-LAT-Port	0-1	0
Login-Service	0-1	0
Login-TCP-Port	0-1	0
-----+-----+		

Attribute Name	Command	
	ACR	ACA
-----+-----+		
NAS-Identifier	0-1	0-1
NAS-IP-Address	0-1	0-1
NAS-IPv6-Address	0-1	0-1
NAS-Port	0-1	0-1
NAS-Port-Id	0-1	0-1
NAS-Port-Type	0-1	0-1
Origin-AAA-Protocol	0-1	0-1
Origin-Host	1	1
Origin-Realm	1	1
Origin-State-Id	0-1	0-1
Originating-Line-Info	0-1	0
Proxy-Info	0+	0+
QoS-Filter-Rule	0+	0
Route-Record	0+	0
Result-Code	0	1
Session-Id	1	1
Service-Type	0-1	0-1
Termination-Cause	0-1	0-1

User-Name	0-1   0-1
-----	-----+-----+

## 6. Unicode Considerations

A number of the AVPs in this RFC use the UTF8String type specified in the Diameter Base protocol [RFC6733]. Implementation differences in Unicode input processing may result in the same Unicode input characters generating different UTF-8 strings that fail to match when compared for equality. This may result in interoperability problems between a network access server and a Diameter server when a UTF-8 string entered locally is compared with one received via Diameter. Many of the uses of UTF8String in this RFC are limited to the 7-bit ASCII-compatible subset of UTF-8 where this class of Unicode string comparison problems does not arise.

Careful preparation of Unicode strings can increase the likelihood that string comparison will work in ways that make sense for typical users throughout the world; [RFC3454] is an example a framework for such Unicode string preparation. The Diameter application specified in this RFC has been deployed with use of Unicode in accordance with [RFC4005], which does not require any Unicode string preparation. As a result, additional requirements for Unicode string preparation in this RFC would not be backwards compatible with existing usage.

The Diameter server and the network access servers that it serves can be assumed to be under common administrative control, and all of the UTF-8 strings involved are part of the configuration of these servers. Therefore administrative interfaces for implementations of this RFC:

- a. SHOULD accept direct UTF-8 input of all configuration strings for AVPs that allow Unicode characters beyond the 7-bit ASCII-compatible subset of Unicode (in addition to any provisions for accepting Unicode characters for processing into UTF-8), and
- b. SHOULD make all such configuration strings available as UTF-8 strings

This functionality enables an administrator who encounters Unicode string comparison problems to copy one instance of a problematic UTF-8 string from one server to the other, after which the two (now identical) copies should compare as expected.

## 7. IANA Considerations

Several of the namespaces used in this document are managed by the Internet Assigned Numbers Authority [IANA], including the AVP Codes

[AVP-Codes], AVP Specific Values [AVP-Vals], Application IDs [App-Ids], Command Codes [Command-Codes] and RADIUS Attribute Values [RADIUSAttrVals].

For the current values allocated, and the policies governing allocation in those namespaces, please see the above-referenced registries.

IANA Note: Please change all the references in the registries listed above that are currently pointing to RFC 4005 to point to this document instead; please change the reference for the value '1' in the "Application IDs" sub-registry of the "Authentication, Authorization, and Accounting (AAA) Parameters" registry to point to this document, as well.

RFC Editor: Please remove both this note and the IANA note above before publication.

## 8. Security Considerations

This document describes the extension of Diameter for the NAS application. 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.

### 8.1. Authentication Considerations

This document does not contain a security protocol but does discuss how PPP authentication protocols can be carried within the Diameter protocol. The PPP authentication protocols described are PAP and CHAP.

The use of PAP SHOULD be discouraged, as it exposes users' passwords to possibly non-trusted entities. However, PAP is also frequently used for use with One-Time Passwords, which do not expose a security risk.

This document also describes how CHAP can be carried within the Diameter protocol, which is required for RADIUS backward compatibility. The CHAP protocol, as used in a RADIUS environment, facilitates authentication replay attacks.

The use of the EAP authentication protocols [RFC4072] can offer better security, given a method suitable for the circumstances.

Depending on the value of the Auth-Request-Type AVP, the Diameter protocol allows authorization-only requests that contain no

authentication information from the client. This capability goes beyond the Call Check capabilities provided by RADIUS (Section 5.6 of [RFC2865]) in that no access decision is requested. As a result, a new session cannot be started as a result of a response to an authorization-only request without introducing a significant security vulnerability.

## 8.2. AVP Considerations

Diameter AVPs often contain security-sensitive data; for example, user passwords and location data, network addresses and cryptographic keys. With the exception of the Configuration-Token (Section 4.4.8), QoS-Filter-Rule (Section 4.4.9) and Tunneling (Section 4.5.1) AVPs, all of the AVPs defined in this document are considered to be security-sensitive.

Diameter messages containing any AVPs considered to be security-sensitive MUST only be sent protected via mutually authenticated TLS or IPsec. In addition, those messages MUST NOT be sent via intermediate nodes unless there is end-to-end security between the originator and recipient or the originator has locally trusted configuration that indicates that end-to-end security is not needed. For example, end-to-end security may not be required in the case where an intermediary node is known to be operated as part of the same administrative domain as the endpoints so that an ability to successfully compromise the intermediary would imply a high probability of being able to compromise the endpoints as well. Note that no end-to-end security mechanism is specified in this document.

## 9. References

### 9.1. Normative References

- [ANITypes] NANPA Number Resource Info, "ANI Assignments", <[http://www.nanpa.com/number\\_resource\\_info/ani\\_ii\\_assignments.html](http://www.nanpa.com/number_resource_info/ani_ii_assignments.html)>.
- [RFC1994] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

- [RFC3162] Aboba, B., Zorn, G., and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001.
- [RFC3516] Nerenberg, L., "IMAP4 Binary Content Extension", RFC 3516, April 2003.
- [RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", RFC 3539, June 2003.
- [RFC5777] Korhonen, J., Tschofenig, H., Arumaithurai, M., Jones, M., and A. Lior, "Traffic Classification and Quality of Service (QoS) Attributes for Diameter", RFC 5777, February 2010.
- [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn, "Diameter Base Protocol", RFC 6733, October 2012.

## 9.2. Informative References

- [ARAP] Apple Computer, "Apple Remote Access Protocol (ARAP) Version 2.0 External Reference Specification", R0612LL/B , September 1994.
- [AVP-Codes] IANA, "IANA AAA AVP Codes Registry", <<http://www.iana.org/assignments/aaa-parameters/aaa-parameters.xml#aaa-parameters-1>>.
- [AVP-Vals] IANA, "IANA AAA AVP Specific Values", <<http://www.iana.org/assignments/aaa-parameters/aaa-parameters.xml#aaa-parameters-2>>.
- [App-Ids] IANA, "IANA AAA Application IDs Registry", <<http://www.iana.org/assignments/aaa-parameters/aaa-parameters.xml#aaa-parameters-1>>.
- [AppleTalk] Sidhu, G., Andrews, R., and A. Oppenheimer, "Inside AppleTalk", Second Edition Apple Computer, 1990.
- [BASE] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.
- [Command-Codes] IANA, "IANA AAA Command Codes Registry", <<http://www.iana.org/assignments/aaa-parameters/aaa-parameters.xml#command-code-rules>>.



- [IANA] IANA, "Internet Assigned Numbers Authority", <<http://www.iana.org/>>.
- [IPX] Novell, Inc., "NetWare System Technical Interface Overview", #883-000780-001, June 1989.
- [ISO.8859-1.1987] International Organization for Standardization, "Information technology - 8-bit single byte coded graphic - character sets - Part 1: Latin alphabet No. 1, JTC1/SC2", ISO Standard 8859-1, 1987.
- [LAT] Digital Equipment Corp., "Local Area Transport (LAT) Specification V5.0", AA-NL26A-TE, June 1989.
- [RADIUSAttrVals] IANA, "IANA Radius Attribute Values Registry", <<http://www.iana.org/assignments/radius-types/radius-types.xml#radius-types-3>>.
- [RFC1334] Lloyd, B. and W. Simpson, "PPP Authentication Protocols", RFC 1334, October 1992.
- [RFC1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.
- [RFC1990] Sklower, K., Lloyd, B., McGregor, G., Carr, D., and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
- [RFC2548] Zorn, G., "Microsoft Vendor-specific RADIUS Attributes", RFC 2548, March 1999.
- [RFC2597] Heinanen, J., Baker, F., Weiss, W., and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, June 1999.
- [RFC2637] Hamzeh, K., Pall, G., Verthein, W., Taarud, J., Little, W., and G. Zorn, "Point-to-Point Tunneling Protocol", RFC 2637, July 1999.
- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

- [RFC2867] Zorn, G., Aboba, B., and D. Mitton, "RADIUS Accounting Modifications for Tunnel Protocol Support", RFC 2867, June 2000.
- [RFC2868] Zorn, G., Leifer, D., Rubens, A., Shriver, J., Holdrege, M., and I. Goyret, "RADIUS Attributes for Tunnel Protocol Support", RFC 2868, June 2000.
- [RFC2869] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.
- [RFC2881] Mitton, D. and M. Beadles, "Network Access Server Requirements Next Generation (NASREQNG) NAS Model", RFC 2881, July 2000.
- [RFC2989] Aboba, B., Calhoun, P., Glass, S., Hiller, T., McCann, P., Shiino, H., Walsh, P., Zorn, G., Dommety, G., Perkins, C., Patil, B., Mitton, D., Manning, S., Beadles, M., Chen, X., Sivalingham, S., Hameed, A., Munson, M., Jacobs, S., Lim, B., Hirschman, B., Hsu, R., Koo, H., Lipford, M., Campbell, E., Xu, Y., Baba, S., and E. Jaques, "Criteria for Evaluating AAA Protocols for Network Access", RFC 2989, November 2000.
- [RFC3169] Beadles, M. and D. Mitton, "Criteria for Evaluating Network Access Server Protocols", RFC 3169, September 2001.
- [RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.
- [RFC3454] , .
- [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, September 2003.
- [RFC3931] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.
- [RFC4072] Eronen, P., Hiller, T., and G. Zorn, "Diameter Extensible Authentication Protocol (EAP) Application", RFC 4072, August 2005.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

## Appendix A. Acknowledgements

### A.1. This Document

The vast majority of the text in this document was taken directly from RFC 4005; the editor owes a debt of gratitude to the authors thereof (especially Dave Mitton, who somehow managed to make nroff paginate the AVP Occurance Tables correctly!).

Thanks (in no particular order) to Jai-Jin Lim, Liu Hans, Sebastien Decugis, Jouni Korhonen, Mark Jones, Hannes Tschofenig, Dave Crocker, David Black, Barry Leiba, Peter Saint-Andre, Stefan Winter and Lionel Morand for their useful reviews and helpful comments.

### A.2. RFC 4005

The authors would like to thank Carl Rigney, Allan C. Rubens, William Allen Simpson, and Steve Willens for their work on the original RADIUS protocol, from which many of the concepts in this specification were derived. Thanks, also, to Carl Rigney for [RFC2866] and [RFC2869]; Ward Willats for [RFC2869]; Glen Zorn, Bernard Aboba, and Dave Mitton for [RFC2867] and [RFC3162]; and Dory Leifer, John Shriver, Matt Holdrege, Allan Rubens, Glen Zorn and Ignacio Goyret for their work on [RFC2868]. This document stole text and concepts from both [RFC2868] and [RFC2869]. Thanks go to Carl Williams for providing IPv6-specific text.

The authors would also like to acknowledge the following people for their contributions in the development of the Diameter protocol: Bernard Aboba, Jari Arkko, William Bulley, Kuntal Chowdhury, Daniel C. Fox, Lol Grant, Nancy Greene, Jeff Hagg, Peter Heitman, Paul Krumviede, Fergal Ladley, Ryan Moats, Victor Muslin, Kenneth Peirce, Sumit Vakil, John R. Vollbrecht, and Jeff Weisberg.

Finally, Pat Calhoun would like to thank Sun Microsystems, as most of the effort put into this document was done while he was in their employ.

### Author's Address

Glen Zorn (editor)  
Network Zen  
227/358 Thanon Sanphawut  
Bang Na, Bangkok 10260  
Thailand

Phone: +66 (0)8-1000-4155  
EMail: glenzorn@gmail.com