Internet Engineering Task Force (IETF)
Request for Comments: 5777
Category: Standards Track
ISSN: 2070-1721

J. Korhonen
H. Tschofenig
Nokia Siemens Networks
M. Arumaithurai
University of Goettingen
M. Jones, Ed.
A. Lior
Bridgewater Systems

February 2010

Traffic Classification and Quality of Service (QoS) Attributes for Diameter

Abstract

This document defines a number of Diameter attribute-value pairs (AVPs) for traffic classification with actions for filtering and Quality of Service (QoS) treatment. These AVPs can be used in existing and future Diameter applications where permitted by the Augmented Backus-Naur Form (ABNF) specification of the respective Diameter command extension policy.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document defines a number of Diameter attribute-value pairs (AVPs) for traffic classification with actions for filtering and Quality of Service (QoS) treatment. These AVPs can be used in existing and future Diameter applications where permitted by the Augmented Backus-Naur Form (ABNF) specification of the respective Diameter command extension policy.

The work on Quality of Service treatment and filtering via Diameter dates back to the base protocol described in RFC 3588 [RFC3588]. The filtering and QoS functionality was provided by the IPFilterRule AVP and the QoSFilterRule AVP. Both AVPs relied on syntax based on the FreeBSD ipfw tool for traffic classification. The functionality of the QoSFilterRule AVP was underspecified in RFC 3588 [RFC3588] and was later updated by RFC 4005 [RFC4005].

As part of the work on updating RFC 3588, the functionality of the IPFilterRule and the QoSFilterRule was revised by the functionality offered by this document with the goals of a uniform and extensible traffic classification mechanism in a native Diameter syntax (instead of the free text previously used). Additionally, an extensible set of actions is provided that offers the ability for filtering and for QoS treatment, whereby the QoS functionality was extended to meet the needs of today’s networking environments.

The QoS-Resources AVP represents a complete rule set with each rule represented by a Filter-Rule AVP. Each rule consists of information for handling conflict resolution, a conditions part and the corresponding actions to be performed if the conditions are satisfied. The AVPs responsible for expressing a condition are defined in Section 4. The capability to match all or a subset of the data traffic is provided. This includes the ability to match on Ethernet specific attributes, which was not possible with the QoS-Filter-Rule AVP. Service differentiation may be based on Ethernet priority bits, a single layer of VLAN-IDs or stacked VLAN-IDs, Logical Link Control (LLC) attributes, MAC addresses, or any combination thereof. The header fields used for Ethernet classification are defined in the IEEE802 series of specifications: [IEEE802.2], [IEEE802.1ad], [IEEE802.1Q], and [IEEE802.1D]. Additionally, time-based conditions can be expressed based on the functionality offered by the attributes in Section 4.2.

The action part of a rule contains the type of traffic treatment and further description regarding QoS-related actions.

The QoS policy rules are defined as Diameter encoded attribute-value pairs (AVPs) described using a modified version of the Augmented Backus-Naur Form (ABNF) (see [RFC3588]). The AVP datatypes are also taken from [RFC3588].

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. Rule Sets and Rules

As mentioned in the introduction, the top-level element is the QoS-Resources AVP that encapsulates one or more Filter-Rule AVPs.

3.1. QoS-Resources AVP

The QoS-Resources AVP (AVP Code 508) is of type Grouped and contains a list of filter policy rules.

```
QoS-Resources ::= < AVP Header: 508 >
                1*{ Filter-Rule }
                * [ AVP ]
```

3.2. Filter-Rule AVP

The Filter-Rule AVP (AVP Code 509) is of type Grouped and defines a specific condition and action combination.

```
                      Filter-Rule ::= < AVP Header: 509 >
                                   [ Filter-Rule-Precedence ]

                                   ; Condition part of a Rule
                                   ; ------------------------

                                   [ Classifier ]
                                 * [ Time-Of-Day-Condition ]

                                   ; Action and Meta-Data
                                   ; --------------------

                                   [ Treatment-Action ]

                                   ; Info about QoS related Actions
                                   ; ------------------------------

                                   [ QoS-Semantics ]
                                   [ QoS-Profile-Template ]
                                   [ QoS-Parameters ]
                                   [ Excess-Treatment ]


                                   ; Extension Point
                                   ; ---------------
                                 * [ AVP ]
```

If the QoS-Profile-Template AVP is not included in the Filter-Rule AVP and the Treatment-Action AVP is set to ’shape’ or ’mark’ then the default setting is assumed, namely, a setting of the Vendor-Id AVP to 0 (for IETF) and the QoS-Profile-Id AVP to zero (0) (for the profile defined in [RFC5624]). Note that the content of the QoS-Parameters are defined in the respective specification defining the QoS parameters. When the Vendor-Id AVP is set to 0 (for IETF) and the

QoS-Profile-Id AVP is set to zero (0), then the AVPs included in the QoS-Parameters AVP are the AVPs defined in [RFC5624].

3.3. Filter-Rule-Precedence AVP

The Filter-Rule-Precedence AVP (AVP Code 510) is of type Unsigned32 and specifies the execution order of the rules expressed in the QoS-Resources AVP. The lower the numerical value of Filter-Rule-Precedence AVP, the higher the rule precedence. Rules with equal precedence MAY be executed in parallel if supported by the Resource Management Function. If the Filter-Rule-Precedence AVP is absent from the Filter-Rule AVP, the rules SHOULD be executed in the order in which they appear in the QoS-Resources AVP.

4. Conditions

This section describes the condition part of a rule. Two condition types are introduced by this document: packet classification conditions represented by the Classifier AVP and time of day conditions represented by the Time-Of-Day-Condition AVP.

If more than one instance of the Time-Of-Day-Condition AVP is present in the Filter-Rule AVP, the current time at rule evaluation MUST be within at least one of the time windows specified in one of the Time-Of-Day-Condition AVPs.

When the Time-Of-Day-Condition AVP and Classifier AVP are present in the same Filter-Rule AVP, both the time of day and packet classification conditions MUST match for the traffic treatment action to be applied.

4.1. Traffic Classifiers

Classifiers are used in many applications to specify how to select a subset of data packets for subsequent treatment as indicated in the action part of a rule. For example, in a QoS application, if a packet matches a classifier then that packet will be treated in accordance with a QoS specification associated with that classifier. Figure 1 shows a typical deployment.

```
                                                           +-----------+
                                                          +-----------+|
       +--------+          +-------------+              +------------+||
       |        |   IN     |             |              |            |||
       |        +--------->|             +------------->|            |||
       |Managed |          | Classifying |              | Unmanaged  |||
       |Terminal|   OUT    | Entity      |              | Terminal   |||
       |        |<---------+             |<-------------+            ||+
       |        |          |             |              |            |+
       +--------+          +-------------+              +------------+
                                  ^
                                  | Classifiers
                                  |
                           +------+------+
                           |             |
                           |     AAA     |
                           |             |
                           +-------------+
```

Figure 1: Example of a Classifier Architecture

The managed terminal, the terminal for which the classifiers are being specified, is located on the left of the Classifying Entity. The unmanaged terminals, the terminals that receive packets from the managed terminal or send packets to the managed terminal, are located to the right side of the Classifying Entity.

The Classifying Entity is responsible for classifying packets that are incoming (IN) from the managed terminal or packets outgoing (OUT) to the managed terminal.

A classifier consists of a group of attributes that specify how to match a packet. Each set of attributes expresses values about aspects of the packet -- typically the packet header. Different protocols therefore would use different attributes.

In general, a classifier consists of the following:

Identifier:

The identifier uniquely identifies this classifier and may be used to reference the classifier from another structure.

From:

Specifies the rule for matching the protocol-specific source address(es) part of the packet.

To:

Specifies the rule for matching the protocol-specific destination address(es) part of the packet.

Protocol:

Specifies the matching protocol of the packet.

Direction:

Specifies whether the classifier is to apply to packets flowing from the managed terminal (IN) or to packets flowing to the managed terminal (OUT) or to packets flowing in both directions.

Options:

Attributes or properties associated with each protocol or layer, or various values specific to the header of the protocol or layer. Options allow matching on those values.

Each protocol type will have a specific set of attributes that can be used to specify a classifier for that protocol. These attributes will be grouped under a grouped AVP called a Classifier AVP.

4.1.1. Classifier AVP

The Classifier AVP (AVP Code 511) is a grouped AVP that consists of a set of attributes that specify how to match a packet.

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

4.1.2. Classifier-ID AVP

The Classifier-ID AVP (AVP Code 512) is of type OctetString and uniquely identifies the classifier. Each application will define the uniqueness scope of this identifier, e.g., unique per terminal or globally unique. Exactly one Classifier-ID AVP MUST be contained within a Classifier AVP.

4.1.3. Protocol AVP

The Protocol AVP (AVP Code 513) is of type Enumerated and specifies the protocol being matched. The attributes included in the Classifier AVP MUST be consistent with the value of the Protocol AVP. Exactly zero or one Protocol AVP may be contained within a Classifier AVP. If the Protocol AVP is omitted from the classifier, then comparison of the protocol of the packet is irrelevant. The values for this AVP are managed by IANA under the Protocol Numbers registry as defined in [RFC2780].

4.1.4. Direction AVP

The Direction AVP (AVP Code 514) is of type Enumerated and specifies in which direction to apply the classifier. The values of the enumeration are "IN","OUT","BOTH". In the "IN" and "BOTH" directions, the From-Spec refers to the address of the managed terminal and the To-Spec refers to the unmanaged terminal. In the "OUT" direction, the From-Spec refers to the unmanaged terminal whereas the To-Spec refers to the managed terminal. If the Direction AVP is omitted, the classifier matches packets flowing in both directions.

Value	Name and Semantic
0	IN - The classifier applies to flows from the managed terminal.
1	OUT - The classifier applies to flows to the managed terminal.
2	BOTH - The classifier applies to flows both to and from the managed terminal.

4.1.5. From-Spec AVP

The From-Spec AVP (AVP Code 515) is a grouped AVP that specifies the Source Specification used to match the packet. Zero or more of these AVPs may appear in the classifier. If this AVP is absent from the classifier, then all packets are matched regardless of the source address. If more than one instance of this AVP appears in the classifier, then the source of the packet can match any From-Spec AVP. The contents of this AVP are protocol specific.

If one instance (or multiple instances) of the IP address AVP (IP-Address, IP-Address-Range, IP-Address-Mask, Use-Assigned-Address) appears in the From-Spec AVP, then the source IP address of the packet MUST match one of the addresses represented by these AVPs.

If more than one instance of the layer 2 address AVPs (MAC-Address, MAC-Address-Mask, EUI64-Address, EUI64-Address-Mask) appears in the From-Spec, then the source layer 2 address of the packet MUST match one of the addresses represented in these AVPs.

If more than one instance of the port AVPs (Port, Port-Range) appears in the From-Spec AVP, then the source port number MUST match one of the port numbers represented in these AVPs.

If the IP address, MAC address, and port AVPs appear in the same From-Spec AVP, then the source packet MUST match all the specifications, i.e., match the IP address AND MAC address AND port number.

```
From-Spec ::= < AVP Header: 515 >
            * [ IP-Address ]
            * [ IP-Address-Range ]
            * [ IP-Address-Mask ]
            * [ MAC-Address ]
            * [ MAC-Address-Mask]
            * [ EUI64-Address ]
            * [ EUI64-Address-Mask]
            * [ Port ]
            * [ Port-Range ]
              [ Negated ]
              [ Use-Assigned-Address ]
            * [ AVP ]
```

4.1.6. To-Spec AVP

The To-Spec AVP (AVP Code 516) is a grouped AVP that specifies the Destination Specification used to match the packet. Zero or more of these AVPs may appear in the classifier. If this AVP is absent from the classifier, then all packets are matched regardless of the destination address. If more than one instance of this AVP appears in the classifier, then the destination of the packet can match any To-Spec AVP. The contents of this AVP are protocol specific.

If one instance (or multiple instances) of the IP address AVP (IP-Address, IP-Address-Range, IP-Address-Mask, Use-Assigned-Address) appears in the To-Spec AVP, then the destination IP address of the packet MUST match one of the addresses represented by these AVPs.

If more than one instance of the layer 2 address AVPs (MAC-Address, MAC-Address-Mask, EUI64-Address, EUI64-Address-Mask) appears in the To-Spec, then the destination layer 2 address of the packet MUST match one of the addresses represented in these AVPs.

If more than one instance of the port AVPs (Port, Port-Range) appears in the To-Spec AVP, then the destination port number MUST match one of the port numbers represented in these AVPs.

If the IP address, MAC address, and port AVPs appear in the same To-Spec AVP, then the destination packet MUST match all the specifications, i.e., match the IP address AND MAC address AND port number.

```
To-Spec ::= < AVP Header: 516 >
          * [ IP-Address ]
          * [ IP-Address-Range ]
          * [ IP-Address-Mask ]
          * [ MAC-Address ]
          * [ MAC-Address-Mask]
          * [ EUI64-Address ]
          * [ EUI64-Address-Mask]
          * [ Port ]
          * [ Port-Range ]
            [ Negated ]
            [ Use-Assigned-Address ]
          * [ AVP ]
```

4.1.7. Source and Destination AVPs

For packet classification, the contents of the From-Spec and To-Spec can contain the AVPs listed in the subsections below.

4.1.7.1. Negated AVP

The Negated AVP (AVP Code 517) is of type Enumerated containing the values of True or False. Exactly zero or one of these AVPs may appear in the From-Spec or To-Spec AVP.

When set to True, the meaning of the match is inverted and the classifier will match addresses other than those specified by the From-Spec or To-Spec AVP. When set to False, or when the Negated AVP is not present, the classifier will match the addresses specified by the From-Spec or To-Spec AVP.

Note that the negation does not impact the port comparisons.

Value	Name
0	False
1	True

4.1.7.2. IP-Address AVP

The IP-Address AVP (AVP Code 518) is of type Address and specifies a single IP address (IPv4 or IPv6) to match.

4.1.7.3. IP-Address-Range AVP

The IP-Address-Range AVP (AVP Code 519) is of type Grouped and specifies an inclusive IP address range.

```
IP-Address-Range ::= < AVP Header: 519 >
                     [ IP-Address-Start ]
                     [ IP-Address-End ]
                   * [ AVP ]
```

If the IP-Address-Start AVP is not included, then the address range starts from the first valid IP address up to and including the specified IP-Address-End address.

If the IP-Address-End AVP is not included, then the address range starts at the address specified by the IP-Address-Start AVP and includes all the remaining valid IP addresses.

For the IP-Address-Range AVP to be valid, the IP-Address-Start AVP MUST contain a value that is less than that of the IP-Address-End AVP.

4.1.7.4. IP-Address-Start AVP

The IP-Address-Start AVP (AVP Code 520) is of type Address and specifies the first IP address (IPv4 or IPv6) of an IP address range.

4.1.7.5. IP-Address-End AVP

The IP-Address-End AVP (AVP Code 521) is of type Address and specifies the last IP address (IPv4 or IPv6) of an address range.

4.1.7.6. IP-Address-Mask AVP

The IP-Address-Mask AVP (AVP Code 522) is of type Grouped and specifies an IP address range using a base IP address and the bit-width of the mask. For example, a range expressed as 192.0.2.0/24 will match all IP addresses from 192.0.2.0 up to and including 192.0.2.255. The bit-width MUST be valid for the type of IP address.

```
IP-Address-Mask ::= < AVP Header: 522 >
                    { IP-Address }
                    { IP-Bit-Mask-Width }
                  * [ AVP ]
```

4.1.7.7. IP-Mask-Bit-Mask-Width AVP

The IP-Bit-Mask-Width AVP (AVP Code 523) is of type Unsigned32. The value specifies the width of an IP address bit mask.

4.1.7.8. MAC-Address AVP

The MAC-Address AVP (AVP Code 524) is of type OctetString and specifies a single layer 2 address in MAC-48 format. The value is a 6-octet encoding of the address as it would appear in the frame header.

4.1.7.9. MAC-Address-Mask AVP

The MAC-Address-Mask AVP (AVP Code 525) is of type Grouped and specifies a set of MAC addresses using a bit mask to indicate the bits of the MAC addresses that must fit to the specified MAC address attribute. For example, a MAC-Address-Mask with the MAC-Address as 00-10-A4-23-00-00 and with a MAC-Address-Mask-Pattern of FF-FF-FF-FF-00-00 will match all MAC addresses from 00-10-A4-23-00-00 up to and including 00-10-A4-23-FF-FF.

Appendix A describes the considerations that should be given to the use of MAC address masks in constructing classifiers.

```
MAC-Address-Mask ::= < AVP Header: 525 >
                     { MAC-Address }
                     { MAC-Address-Mask-Pattern }
                   * [ AVP ]
```

4.1.7.10. MAC-Address-Mask-Pattern AVP

The MAC-Address-Mask-Pattern AVP (AVP Code 526) is of type OctetString. The value is 6 octets specifying the bit positions of a MAC address that are taken for matching.

4.1.7.11. EUI64-Address AVP

The EUI64-Address AVP (AVP Code 527) is of type OctetString and specifies a single layer 2 address in EUI-64 format. The value is an 8-octet encoding of the address as it would appear in the frame header.

4.1.7.12. EUI64-Address-Mask AVP

The EUI64-Address-Mask AVP (AVP Code 528) is of type Grouped and specifies a set of EUI64 addresses using a bit mask to indicate the bits of the EUI64 addresses that must fit to the specified EUI64 address attribute. For example, a EUI64-Address-Mask with the EUI64-Address as 00-10-A4-FF-FE-23-00-00 and with a EUI64-Address-Mask-Pattern of FF-FF-FF-FF-FF-FF-00-00 will match all EUI64 addresses from 00-10-A4-FF-FE-23-00-00 up to and including 00-10-A4-FF-FE-23-FF-FF.

Appendix A describes the considerations that should be given to the use of EUI64 address masks in constructing classifiers.

```
EUI64-Address-Mask ::= < AVP Header: 528 >
                       { EUI64-Address }
                       { EUI64-Address-Mask-Pattern }
                     * [ AVP ]
```

4.1.7.13. EUI64-Address-Mask-Pattern AVP

The EUI64-Address-Mask-Pattern AVP (AVP Code 529) is of type OctetString. The value is 8 octets specifying the bit positions of a EUI64 address that are taken for matching.

4.1.7.14. Port AVP

The Port AVP (AVP Code 530) is of type Integer32 in the range of 0 to 65535 and specifies port numbers to match. The type of port is indicated by the value of the Protocol AVP; i.e., if Protocol AVP value is 6 (TCP), then the Port AVP represents a TCP port.

4.1.7.15. Port-Range AVP

The Port-Range AVP (AVP Code 531) is of type Grouped and specifies an inclusive range of ports. The type of the ports is indicated by the value of the Protocol AVP; i.e., if Protocol AVP value is 6 (TCP), then the Port-Range AVP represents an inclusive range of TCP ports.

```
Port-Range ::= < AVP Header: 531 >
               [ Port-Start ]
               [ Port-End ]
             * [ AVP ]
```

If the Port-Start AVP is omitted, then port 0 is assumed. If the Port-End AVP is omitted, then port 65535 is assumed.

4.1.7.16. Port-Start AVP

The Port-Start AVP (AVP Code 532) is of type Integer32 and specifies the first port number of an IP port range.

4.1.7.17. Port-End AVP

The Port-End AVP (AVP Code 533) is of type Integer32 and specifies the last port number of an IP port range.

4.1.7.18. Use-Assigned-Address AVP

In some scenarios, the AAA does not know the IP address assigned to the managed terminal at the time that the classifier is sent to the Classifying Entity. The Use-Assigned-Address AVP (AVP Code 534) is of type Enumerated containing the values of True or False. When present and set to True, it represents the IP address assigned to the managed terminal.

Value	Name
0	False
1	True

4.1.8. Header Option AVPs

The Classifier AVP may contain one or more of the following AVPs to match on the various possible IP, TCP, or ICMP header options.

4.1.8.1. Diffserv-Code-Point AVP

The Diffserv-Code-Point AVP (AVP Code 535) is of type Enumerated and specifies the Differentiated Services Field Codepoints to match in the IP header. The values are managed by IANA under the Differentiated Services Field Codepoints registry as defined in [RFC2474].

4.1.8.2. Fragmentation-Flag AVP

The Fragmentation-Flag AVP (AVP Code 536) is of type Enumerated and specifies the packet fragmentation flags to match in the IP header.

Value	Name and Semantic
0	Don’t Fragment (DF)
1	More Fragments (MF)

4.1.8.3. IP-Option AVP

The IP-Option AVP (AVP Code 537) is of type Grouped and specifies an IP header option that must be matched.

```
IP-Option ::= < AVP Header: 537 >
                { IP-Option-Type }
              * [ IP-Option-Value ]
                [ Negated ]
              * [ AVP ]
```

If one or more IP-Option-Value AVPs are present, one of the values MUST match the value in the IP header option. If the IP-Option-Value AVP is absent, the option type MUST be present in the IP header but the value is wild carded.

The Negated AVP is used in conjunction with the IP-Option-Value AVPs to specify IP header options that do not match specific values. The Negated AVP is used without the IP-Option-Value AVP to specify IP headers that do not contain the option type.

4.1.8.4. IP-Option-Type AVP

The IP-Option-Type AVP (AVP Code 538) is of type Enumerated and the values are managed by IANA under the IP Option Numbers registry as defined in [RFC2780].

4.1.8.5. IP-Option-Value AVP

The IP-Option-Value AVP (AVP Code 539) is of type OctetString and contains the option value that must be matched.

4.1.8.6. TCP-Option AVP

The TCP-Option AVP (AVP Code 540) is of type Grouped and specifies a TCP header option that must be matched.

```
TCP-Option ::= < AVP Header: 540 >
                 { TCP-Option-Type }
               * [ TCP-Option-Value ]
                 [ Negated ]
               * [ AVP ]
```

If one or more TCP-Option-Value AVPs are present, one of the values MUST match the value in the TCP header option. If the TCP-Option-Value AVP is absent, the option type MUST be present in the TCP header but the value is wild carded.

The Negated AVP is used in conjunction that the TCP-Option-Value AVPs to specify TCP header options that do not match specific values. The Negated AVP is used without the TCP-Option-Value AVP to specify TCP headers that do not contain the option type.

4.1.8.7. TCP-Option-Type AVP

The TCP-Option-Type AVP (AVP Code 541) is of type Enumerated and the values are managed by IANA under the TCP Option Numbers registry as defined in [RFC2780].

4.1.8.8. TCP-Option-Value AVP

The TCP-Option-Value AVP (AVP Code 542) is of type OctetString and contains the option value that must be matched.

4.1.8.9. TCP-Flags AVP

The TCP-Flags AVP (AVP Code 543) is of type Grouped and specifies a set of TCP control flags that must be matched.

```
TCP-Flags ::= < AVP Header: 543 >
                { TCP-Flag-Type }
                [ Negated ]
              * [ AVP ]
```

If the Negated AVP is not present or present but set to False, the TCP-Flag-Type AVP specifies which flags MUST be set. If the Negated AVP is set to True, the TCP-Flag-Type AVP specifies which flags MUST be cleared.

4.1.8.10. TCP-Flag-Type AVP

The TCP-Flag-Type AVP (AVP Code 544) is of type Unsigned32 and specifies the TCP control flag types that must be matched. The first 16 bits match the TCP header format defined in [RFC3168], and the subsequent 16 bits are unused. Within the first 16 bits, bits 0 to 3 are unused and bits 4 to 15 are managed by IANA under the TCP Header Flag registry as defined in [RFC3168].

4.1.8.11. ICMP-Type

The ICMP-Type AVP (AVP Code 545) is of type Grouped and specifies an ICMP message type that must be matched.

```
ICMP-Type ::= < AVP Header: 545 >
                { ICMP-Type-Number }
              * [ ICMP-Code ]
                [ Negated ]
              * [ AVP ]
```

If the ICMP-Code AVP is present, the value MUST match that in the ICMP header. If the ICMP-Code AVP is absent, the ICMP type MUST be present in the ICMP header but the code is wild carded.

The Negated AVP is used in conjunction with the ICMP-Code AVPs to specify ICMP codes that do not match specific values. The Negated AVP is used without the ICMP-Code AVP to specify ICMP headers that do not contain the ICMP type. As such, the Negated AVP feature applies to ICMP-Code AVP if the ICMP-Code AVP is present. If the ICMP-Code AVP is absent, the Negated AVP feature applies to the ICMP-Type-Number.

4.1.8.12. ICMP-Type-Number AVP

The ICMP-Type-Number AVP (AVP Code 546) is of type Enumerated and the values are managed by IANA under the ICMP Type Numbers registry as defined in [RFC2780].

4.1.8.13. ICMP-Code AVP

The ICMP-Code AVP (AVP Code 547) is of type Enumerated and the values are managed by IANA under the ICMP Type Numbers registry as defined in [RFC2780].

4.1.8.14. ETH-Option AVP

The ETH-Option AVP (AVP Code 548) is of type Grouped and specifies Ethernet specific attributes.

```
ETH-Option ::= < AVP Header: 548 >
               { ETH-Proto-Type }
             * [ VLAN-ID-Range ]
             * [ User-Priority-Range ]
             * [ AVP ]
```

4.1.8.15. ETH-Proto-Type AVP

The Eth-Proto-Type AVP (AVP Code 549) is of type Grouped and specifies the encapsulated protocol type. ETH-Ether-Type and ETH-SAP are mutually exclusive.

```
ETH-Proto-Type ::= < AVP Header: 549 >
                 * [ ETH-Ether-Type ]
                 * [ ETH-SAP ]
                 * [ AVP ]
```

4.1.8.16. ETH-Ether-Type AVP

The ETH-Ether-Type AVP (AVP Code 550) is of type OctetString. The value is a double octet that contains the value of the Ethertype field in the packet to match. This AVP MAY be present in the case of Digital, Intel, and Xerox (DIX) or if the Subnetwork Access Protocol (SNAP) is present at 802.2, but the ETH-SAP AVP MUST NOT be present in this case.

4.1.8.17. ETH-SAP AVP

The ETH-SAP AVP (AVP Code 551) is of type OctetString. The value is a double octet representing the 802.2 SAP as specified in [IEEE802.2]. The first octet contains the Destination Service Access Point (DSAP) and the second the Source Service Access Point (SSAP).

4.1.8.18. VLAN-ID-Range AVP

The VLAN-ID-Range AVP (AVP Code 552) is of type Grouped and specifies the VLAN range to match. VLAN identities are specified either by a single VLAN-ID according to [IEEE802.1Q] or by a combination of Customer and Service VLAN-IDs according to [IEEE802.1ad].

The single VLAN-ID is represented by the C-VID-Start and C-VID-End AVPs, and the S-VID-Start and S-VID-End AVPs SHALL be omitted in this case. If the VLAN-ID-Range AVP is omitted from the classifier, then comparison of the VLAN identity of the packet is irrelevant.

```
VLAN-ID-Range ::= < AVP Header: 552 >
                  [ S-VID-Start ]
                  [ S-VID-End ]
                  [ C-VID-Start ]
                  [ C-VID-End ]
                * [ AVP ]
```

The following is the list of possible combinations of the S-VID-Start and S-VID-End AVPs and their inference:

- If S-VID-Start AVP is present but the S-VID-End AVP is absent, the S-VID-Start AVP value MUST equal the value of the IEEE 802.1ad S-VID bits specified in [IEEE802.1ad] for a successful match.

- If S-VID-Start AVP is absent but the S-VID-End AVP is present, the S-VID-End AVP value MUST equal the value of the IEEE 802.1ad S-VID bits for a successful match.

- If both S-VID-Start and S-VID-End AVPs are present and their values are equal, the S-VID-Start AVP value MUST equal the value of the IEEE 802.1ad S-VID bits for a successful match.

- If both S-VID-Start and S-VID-End AVPs are present and the value of S-VID-End AVP is greater than the value of the S-VID-Start AVP, the value of the IEEE 802.1ad S-VID bits MUST be greater than or equal to the S-VID-Start AVP value and less than or equal to the S-VID-End AVP value for a successful match. If the S-VID-Start and S-VID-End AVPs are specified, then Ethernet packets without IEEE 802.1ad encapsulation MUST NOT match this classifier.

- If the S-VID-Start and S-VID-End AVPs are omitted, then existence of IEEE802.1ad encapsulation or comparison of the IEEE 802.1ad S-VID bits is irrelevant for this classifier.

The following is the list of possible combinations of the C-VID-Start and C-VID-End AVPs and their inference:

- If C-VID-Start AVP is present but the C-VID-End AVP is absent, the C-VID-Start AVP value MUST equal the value of the IEEE 802.1ad C-VID bits specified in [IEEE802.1ad] or the IEEE 802.1Q VLAN-ID bits specified in [IEEE802.1Q] for a successful match.

- If C-VID-Start AVP is absent but the C-VID-End AVP is present, the C-VID-End AVP value MUST equal the value of the IEEE 802.1ad C-VID bits or the IEEE 802.1Q VLAN-ID bits for a successful match.

- If both C-VID-Start and C-VID-End AVPs are present and their values are equal, the C-VID-Start AVP value MUST equal the value of the IEEE 802.1ad C-VID bits or the IEEE 802.1Q VLAN-ID bits for a successful match.

- If both C-VID-Start and C-VID-End AVPs are present and the value of C-VID-End AVP is greater than the value of the C-VID-Start AVP, the value of the IEEE 802.1ad C-VID bits or the IEEE 802.1Q VLAN-ID bits MUST be greater than or equal to the C-VID-Start AVP value and less than or equal to the C-VID-End AVP value for a successful match. If the C-VID-Start and C-VID-End AVPs are specified, then Ethernet packets without IEEE 802.1ad or IEEE 802.1Q encapsulation MUST NOT match this classifier.

- If the C-VID-Start and C-VID-End AVPs are omitted, the comparison of the IEEE 802.1ad C-VID bits or IEEE 802.1Q VLAN-ID bits for this classifier is irrelevant.

4.1.8.19. S-VID-Start AVP

The S-VID-Start AVP (AVP Code 553) is of type Unsigned32. The value MUST be in the range from 0 to 4095. The value of this AVP specifies the start value of the range of S-VID VLAN-IDs to be matched.

4.1.8.20. S-VID-End AVP

The S-VID-End AVP (AVP Code 554) is of type Unsigned32. The value MUST be in the range from 0 to 4095. The value of this AVP specifies the end value of the range of S-VID VLAN-IDs to be matched.

4.1.8.21. C-VID-Start AVP

The C-VID-Start AVP (AVP Code 555) is of type Unsigned32. The value MUST be in the range from 0 to 4095. The value of this AVP specifies the start value of the range of C-VID VLAN-IDs to be matched.

4.1.8.22. C-VID-End AVP

The C-VID-End AVP (AVP Code 556) is of type Unsigned32. The value MUST be in the range from 0 to 4095. The value of this AVP specifies the end value of the range of C-VID VLAN-IDs to be matched.

4.1.8.23. User-Priority-Range AVP

The User-Priority-Range AVP (AVP Code 557) is of type Grouped and specifies an inclusive range to match the user_priority parameter specified in [IEEE802.1D]. An Ethernet packet containing the user_priority parameter matches this classifier if the value is greater than or equal to Low-User-Priority and less than or equal to High-User-Priority. If this AVP is omitted, then comparison of the IEEE 802.1D user_priority parameter for this classifier is irrelevant.

```
User-Priority-Range ::= < AVP Header: 557 >
                      * [ Low-User-Priority ]
                      * [ High-User-Priority ]
                      * [ AVP ]
```

4.1.8.24. Low-User-Priority AVP

The Low-User-Priority AVP (AVP Code 558) is of type Unsigned32. The value MUST be in the range from 0 to 7.

4.1.8.25. High-User-Priority AVP

The High-User-Priority AVP (AVP Code 559) is of type Unsigned32. The value MUST be in the range from 0 to 7.

4.2. Time Of Day AVPs

In many QoS applications, the QoS specification applied to the traffic flow is conditional upon the time of day when the flow was observed. The following sections define AVPs that can be used to express one or more time windows that determine when a traffic treatment action is applicable to a traffic flow.

4.2.1. Time-Of-Day-Condition AVP

The Time-Of-Day-Condition AVP (AVP Code 560) is of type Grouped and specifies one or more time windows.

```
Time-Of-Day-Condition ::= < AVP Header: 560 >
                          [ Time-Of-Day-Start ]
                          [ Time-Of-Day-End ]
                          [ Day-Of-Week-Mask ]
                          [ Day-Of-Month-Mask ]
                          [ Month-Of-Year-Mask ]
                          [ Absolute-Start-Time ]
                          [ Absolute-End-Time ]
                          [ Timezone-Flag ]
                        * [ AVP ]
```

For example, a time window for 9 a.m. to 5 p.m. (local time) from Monday to Friday would be expressed as:

```
Time-Of-Day-Condition = {
    Time-Of-Day-Start = 32400;
    Time-Of-Day-End = 61200;
    Day-Of-Week-Mask =
        ( MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY );
    Timezone-Flag = LOCAL;
}
```

4.2.2. Time-Of-Day-Start AVP

The Time-Of-Day-Start AVP (AVP Code 561) is of type Unsigned32. The value MUST be in the range from 0 to 86400. The value of this AVP specifies the start of an inclusive time window expressed as the offset in seconds from midnight. If this AVP is absent from the Time-Of-Day-Condition AVP, the time window starts at midnight.

4.2.3. Time-Of-Day-End AVP

The Time-Of-Day-End AVP (AVP Code 562) is of type Unsigned32. The value MUST be in the range from 1 to 86400. The value of this AVP specifies the end of an inclusive time window expressed as the offset in seconds from midnight. If this AVP is absent from the Time-Of-Day-Condition AVP, the time window ends one second before midnight.

4.2.4. Day-Of-Week-Mask AVP

The Day-Of-Week-Mask AVP (AVP Code 563) is of type Unsigned32. The value is a bit mask that specifies the day of the week for the time window to match. This document specifies the following bits:

Bit	Name
0	SUNDAY
1	MONDAY
2	TUESDAY
3	WEDNESDAY
4	THURSDAY
5	FRIDAY
6	SATURDAY

The bit MUST be set for the time window to match on the corresponding day of the week. Bit 0 is the least significant bit and unused bits MUST be cleared. If this AVP is absent from the Time-Of-Day-Condition AVP, the time windows match on all days of the week.

4.2.5. Day-Of-Month-Mask AVP

The Day-Of-Month AVP (AVP Code 564) is of type Unsigned32. The value MUST be in the range from 0 to 2147483647. The value is a bit mask that specifies the days of the month where bit 0 represents the first day of the month through to bit 30, which represents the last day of the month. The bit MUST be set for the time window to match on the corresponding day of the month. Bit 0 is the least significant bit and unused bits MUST be cleared. If this AVP is absent from the Time-Of-Day-Condition AVP, the time windows match on all days of the month.

4.2.6. Month-Of-Year-Mask AVP

The Month-Of-Year-Mask AVP (AVP Code 565) is of type Unsigned32. The value is a bit mask that specifies the months of the year for the time window to match. This document specifies the following bits:

Bit	Name
0	JANUARY
1	FEBRUARY
2	MARCH
3	APRIL
4	MAY
5	JUNE
6	JULY
7	AUGUST
8	SEPTEMBER
9	OCTOBER
10	NOVEMBER
11	DECEMBER

The bit MUST be set for the time window to match on the corresponding month of the year. Bit 0 is the least significant bit and unused bits MUST be cleared. If this AVP is absent from the Time-Of-Day-Condition AVP, the time windows match during all months of the year.

4.2.7. Absolute-Start-Time AVP

The Absolute-Start-Time AVP (AVP Code 566) is of type Time. The value of this AVP specifies the time in seconds since January 1, 1900, 00:00 UTC when the time window starts. If this AVP is absent from the Time-Of-Day-Condition AVP, the time window starts on January 1, 1900, 00:00 UTC.

4.2.8. Absolute-Start-Fractional-Seconds AVP

The Absolute-Start-Fractional-Seconds AVP (AVP Code 567) is of type Unsigned32. The value specifies the fractional seconds that are added to Absolute-Start-Time value in order to determine when the time window starts. If this AVP is absent from the Time-Of-Day-Condition AVP, then the fractional seconds are assumed to be zero.

4.2.9. Absolute-End-Time AVP

The Time-Of-Day-End AVP (AVP Code 568) is of type Time. The value of this AVP specifies the time in seconds since January 1, 1900, 00:00 UTC when the time window ends. If this AVP is absent from the Time-Of-Day-Condition AVP, then the time window is open-ended.

4.2.10. Absolute-End-Fractional-Seconds AVP

The Absolute-End-Fractional-Seconds AVP (AVP Code 569) is of type Unsigned32. The value specifies the fractional seconds that are added to Absolute-End-Time value in order to determine when the time window ends. If this AVP is absent from the Time-Of-Day-Condition AVP, then the fractional seconds are assumed to be zero.

4.2.11. Timezone-Flag AVP

The Timezone-Flag AVP (AVP Code 570) is of type Enumerated and indicates whether the time windows are specified in UTC, local time, at the managed terminal or as an offset from UTC. If this AVP is absent from the Time-Of-Day-Condition AVP, then the time windows are in UTC.

This document defines the following values:

Value	Name and Semantic
0	UTC - The time windows are expressed in UTC.
1	LOCAL - The time windows are expressed in local time at the managed terminal.
2	OFFSET - The time windows are expressed as an offset from UTC (see Timezone-Offset AVP).

4.2.12. Timezone-Offset AVP

The Timezone-Offset AVP (AVP Code 571) is of type Integer32. The value of this AVP MUST be in the range from -43200 to 43200. It specifies the offset in seconds from UTC that was used to express Time-Of-Day-Start, Time-Of-Day-End, Day-Of-Week-Mask, Day-Of-Month-Mask, and Month-Of-Year-Mask AVPs. This AVP MUST be present if the Timezone-Flag AVP is set to OFFSET.

5. Actions

This section defines the actions associated with a rule.

5.1. Treatment-Action AVP

The Treatment-Action AVP (AVP Code 572) is of type Enumerated and lists the actions that are associated with the condition part of a rule. The following actions are defined in this document:

0: drop
1: shape
2: mark
3: permit

drop:

This action indicates that the respective traffic MUST be dropped.

shape:

[RFC2475] describes shaping as "the process of delaying packets within a traffic stream to cause it to conform to some defined traffic profile". When the action is set to ’shape’, the QoS-Parameters AVP SHALL contain QoS information AVPs, such as the TMOD-1 and Bandwidth AVPs [RFC5624], that indicate how to shape the traffic described by the condition part of the rule.

mark:

[RFC2475] describes marking as "the process of setting the DS codepoint in a packet based on defined rules". When the action is set to 'mark', the QoS-Parameters AVP SHALL contain QoS information AVPs, such as the PHB-Class AVP [RFC5624], that indicate the Diffserv marking to be applied to the traffic described by the condition part of the rule.

permit:

The 'permit' action is the counterpart to the 'drop' action used to allow traffic that matches the condition part of a rule to bypass.

[RFC2475] also describes an action called 'policing' as "the process of discarding packets (by a dropper) within a traffic stream in accordance with the state of a corresponding meter enforcing a traffic profile". This behavior is modeled in the Filter-Rule through the inclusion of the Excess-Treatment AVP containing a Treatment-Action AVP set to 'drop'.

Further action values can be registered, as described in Section 10.3.

5.2. QoS-Profile-Id AVP

The QoS-Profile-Id AVP (AVP Code 573) is of type Unsigned32 and contains a QoS profile template identifier. An initial QoS profile template is defined with value of 0 and can be found in [RFC5624]. The registry for the QoS profile templates is created with the same document.

5.3. QoS-Profile-Template AVP

The QoS-Profile-Template AVP (AVP Code 574) is of type Grouped and defines the namespace of the QoS profile (indicated in the Vendor-ID AVP) followed by the specific value for the profile.

The Vendor-Id AVP contains a 32-bit IANA Private Enterprise Number (PEN), and the QoS-Profile-Id AVP contains the template identifier assigned by the vendor. The vendor identifier of zero (0) is used for the IETF.

```
   QoS-Profile-Template ::= < AVP Header: 574 >
                             { Vendor-Id }
                             { QoS-Profile-Id }
                           * [ AVP ]
```

5.4. QoS-Semantics

The QoS-Semantics AVP (AVP Code 575) is of type Enumerated and provides the semantics for the QoS-Profile-Template and QoS-Parameters AVPs in the Filter-Rule AVP.

This document defines the following values:

(0): QoS-Desired
(1): QoS-Available
(2): QoS-Delivered
(3): Minimum-QoS
(4): QoS-Authorized

The semantics of the QoS parameters depend on the information provided in the list above. The semantics of the different values are as follows:

Object Type	Direction	Semantic
QoS-Desired	C->S	Client requests authorization of the indicated QoS.
QoS-Desired	C<-S	NA
QoS-Available	C->S	Admission Control at client indicates that this QoS is available. (note 1)
QoS-Available	C<-S	Admission Control at server indicates that this QoS is available. (note 2)
QoS-Delivered	C->S	Client is reporting the actual QoS delivered to the terminal.
QoS-Delivered	C<-S	NA
Minimum-QoS	C->S	Client is not interested in authorizing QoS that is lower than the indicated QoS.
Minimum-QoS	C<-S	Client must not provide QoS guarantees lower than the indicated QoS.
QoS-Authorized	C->S	NA
QoS-Authorized	C<-S	Server authorizes the indicated QoS.

Legend:

C: Diameter client
S: Diameter server
NA: Not applicable to this document;
    no semantic defined in this specification

Notes:

(1) QoS-Available in this direction indicates to the server that any QoS-Authorized or Minimum-QoS must be less than this indicated QoS.

(2) QoS-Available in this direction is only useful when the AAA server performs admission control and knows about the resources in the network.

5.5. QoS-Parameters AVP

The QoS-Parameters AVP (AVP Code 576) is of type Grouped and contains Quality of Service parameters. These parameters are defined in separate documents and depend on the indicated QoS profile template of the QoS-Profile-Template AVP. For an initial QoS parameter specification, see [RFC5624].

```
QoS-Parameters  ::= < AVP Header: 576 >
                     * [ AVP ]
```

5.6. Excess-Treatment AVP

The Excess-Treatment AVP (AVP Code 577) is of type Grouped and indicates how out-of-profile traffic, i.e., traffic not covered by the original QoS-Profile-Template and QoS-Parameters AVPs, is treated. The additional Treatment-Action, QoS-Profile-Template, and QoS-Parameters AVPs carried inside the Excess-Treatment AVP provide information about the QoS treatment of the excess traffic. In case the Excess-Treatment AVP is absent, then the treatment of the out-of-profile traffic is left to the discretion of the node performing QoS treatment.

```
Excess-Treatment ::= < AVP Header: 577 >
                     { Treatment-Action }
                     [ QoS-Profile-Template ]
                     [ QoS-Parameters ]
                   * [ AVP ]
```

6. QoS Capability Indication

The QoS-Capability AVP (AVP Code 578) is of type Grouped and contains a list of supported Quality of Service profile templates (and therefore the support of the respective parameter AVPs).

The QoS-Capability AVP may be used for a simple announcement of the QoS capabilities and QoS profiles supported by a peer. It may also be used to negotiate a mutually supported set of QoS capabilities and

QoS profiles between two peers. In such a case, handling of failed negotiations is application and/or deployment specific.

```
QoS-Capability ::= < AVP Header: 578 >
                 1*{ QoS-Profile-Template }
                  * [ AVP ]
```

The QoS-Profile-Template AVP is defined in Section 5.3.

7. Examples

This section shows a number of signaling flows where QoS negotiation and authorization are part of the conventional NASREQ, EAP, or Credit Control applications message exchanges. The signaling flows for the Diameter QoS Application are described in [DIAMETER-QOS].

7.1. Diameter EAP with QoS Information

Figure 2 shows a simple signaling flow where a Network Access Server (NAS) (Diameter Client) announces its QoS awareness and capabilities included into the DER message and as part of the access authentication procedure. Upon completion of the EAP exchange, the Diameter server provides a pre-provisioned QoS profile with the QoS-Semantics in the Filter-Rule AVP set to ’QoS-Authorized’, to the NAS in the final Diameter-EAP-Answer (DEA) message.

```
    End                           Diameter                      Diameter
    Host                           Client                         Server
     |                               |                                |
     |        (initiate EAP)         |                                |
     |<----------------------------->|                                |
     |                               | Diameter-EAP-Request           |
     |                               | EAP-Payload(EAP Start)         |
     |                               | QoS-Capability                 |
     |                               |------------------------------->|
     |                               |                                |
     |                               |            Diameter-EAP-Answer |
     |                          Result-Code=DIAMETER_MULTI_ROUND_AUTH |
     |                               |    EAP-Payload(EAP Request #1) |
     |                               |<-------------------------------|
     |         EAP Request(Identity) |                                |
     |<------------------------------|                                |
     :                               :                                :
     :                    <<<more message exchanges>>>                :
     :                               :                                :
     |                               |                                |
     | EAP Response #N               |                                |
     |------------------------------>|                                |
     |                               | Diameter-EAP-Request           |
     |                               | EAP-Payload(EAP Response #N)   |
     |                               |------------------------------->|
     |                               |                                |
     |                               |            Diameter-EAP-Answer |
     |                               |   Result-Code=DIAMETER_SUCCESS |
     |                               |       EAP-Payload(EAP Success) |
     |                               |           (authorization AVPs) |
     |                               |  QoS-Resources(QoS-Authorized) |
     |                               |<-------------------------------|
     |                               |                                |
     |                   EAP Success |                                |
     |<------------------------------|                                |
     |                               |                                |
```

Figure 2: Example of a Diameter EAP Enhanced with QoS Information

7.2. Diameter NASREQ with QoS Information

Figure 3 shows a similar pre-provisioned QoS signaling as in Figure 2 but using the NASREQ application instead of EAP application.

```
      End                                             Diameter
      Host                NAS                           Server
       |                  |                              |
       |  Start Network   |                              |
       |  Attachment      |                              |
       |<---------------->|                              |
       |                  |                              |
       |                  |AA-Request                    |
       |                  |NASREQ-Payload                |
       |                  |QoS-Capability                |
       |                  +----------------------------->|
       |                  |                              |
       |                  |                     AA-Answer|
       |            Result-Code=DIAMETER_MULTI_ROUND_AUTH|
       |                NASREQ-Payload(NASREQ Request #1)|
       |                  |<-----------------------------+
       |                  |                              |
       | Request          |                              |
       |<-----------------+                              |
       |                  |                              |
       :                  :                              :
       :           <<<more message exchanges>>>          :
       :                  :                              :
       | Response #N      |                              |
       +----------------->|                              |
       |                  |                              |
       |                  |AA-Request                    |
       |                  |NASREQ-Payload ( Response #N )|
       |                  +----------------------------->|
       |                  |                              |
       |                  |                     AA-Answer|
       |                  |  Result-Code=DIAMETER_SUCCESS|
       |                  |          (authorization AVPs)|
       |                  | QoS-Resources(QoS-Authorized)|
       |                  |<-----------------------------+
       |                  |                              |
       | Success          |                              |
       |<-----------------+                              |
       |                  |                              |
```

Figure 3: Example of a Diameter NASREQ Enhanced with QoS Information

7.3. QoS Authorization

Figure 4 shows an example of authorization-only QoS signaling as part of the NASREQ message exchange. The NAS provides the Diameter server with the "QoS-Desired" QoS-Semantics AVP included in the QoS-Resources AVP. The Diameter server then either authorizes the indicated QoS or rejects the request and informs the NAS about the result. In this scenario, the NAS does not need to include the QoS-Capability AVP in the AAR message as the QoS-Resources AVP implicitly does the same and also the NAS is authorizing a specific QoS profile, not a pre-provisioned one.

```
       End                                            Diameter
       Host              NAS                           Server
        |                  |                              |
        |                  |                              |
        |  QoS Request     |                              |
        +----------------->|                              |
        |                  |                              |
        |                  |AA-Request                    |
        |                  |Auth-Request-Type=AUTHORIZE_ONLY
        |                  |NASREQ-Payload                |
        |                  |QoS-Resources(QoS-Desired)    |
        |                  +----------------------------->|
        |                  |                              |
        |                  |                     AA-Answer|
        |                  |       NASREQ-Payload(Success)|
        |                  | QoS-Resources(QoS-Authorized)|
        |                  |<-----------------------------+
        |  Accept          |                              |
        |<-----------------+                              |
        |                  |                              |
        |                  |                              |
        |                  |                              |
```

Figure 4: Example of an Authorization-Only Message Flow

7.4. Diameter Server Initiated Re-Authorization of QoS

Figure 5 shows a message exchange for a Diameter server-initiated QoS re-authorization procedure. The Diameter server sends the NAS a Re-Auth Request (RAR) message requesting re-authorization for an existing session and the NAS acknowledges it with a RAA message. The NAS is aware of its existing QoS profile and information for the ongoing session that the Diameter server requested for re-authorization. Thus, the NAS must initiate re-authorization of the existing QoS profile. The re-authorization procedure is the same as in Figure 4.

```
      End                                               Diameter
      Host                NAS                            Server
       |                  |                              |
       |                  |                              |
       :                  :                              :
       :          <<<Initial Message Exchanges>>>        :
       :                  :                              :
       |                  |                              |
       |                  |                   RA-Request |
       |                  |<-----------------------------+
       |                  |                              |
       |                  |RA-Answer                     |
       |                  |Result-Code=DIAMETER_SUCCESS  |
       |                  +----------------------------->|
       |                  |                              |
       |                  |                              |
       |                  |AA-Request                    |
       |                  |NASREQ-Payload                |
       |                  |Auth-Request-Type=AUTHORIZE_ONLY
       |                  |QoS-Resources(QoS-Desired)    |
       |                  +----------------------------->|
       |                  |                              |
       |                  |                     AA-Answer|
       |                  |  Result-Code=DIAMETER_SUCCESS|
       |                  |          (authorization AVPs)|
       |                  | QoS-Resources(QoS-Authorized)|
       |                  |<-----------------------------+
       |                  |                              |
```

Figure 5: Example of a Server-Initiated Re-Authorization Procedure

7.5. Diameter Credit Control (CC) with QoS Information

In this example, the CC client includes a QoS authorization request (QoS-Semantics=QoS-Desired) in the initial Credit Control Request (CCR). The CC server responds with a Credit Control Answer (CCA), which includes the granted resources with an authorized QoS definition (QoS-Semantics=QoS-Authorized) and the CC client proceeds to deliver service with the specified QoS.

At the end of service, the CC client reports the units used and the QoS level at which those units were delivered (QoS-Semantics=QoS-Delivered). The end of service could occur because the credit resources granted to the user were exhausted or the service was been successfully delivered or the service was terminated, e.g., because the Service Element could not deliver the service at the authorized QoS level.

```
                           Service Element
     End User            (CC Client)                        CC Server
        |                     |                                  |
        |(1) Service Request  |                                  |
        |-------------------->|                                  |
        |                     |(2) CCR (Initial,                 |
        |                     |    QoS-Resources(QoS-Desired))   |
        |                     |--------------------------------->|
        |                     |(3) CCA (Granted-Units,           |
        |                     |    QoS-Resources(QoS-Authorized))|
        |                     |<---------------------------------|
        |(4) Service Delivery |                                  |
        |<------------------->|                                  |
        |                     |                                  |
        |(5) End of Service   |                                  |
        |-------------------->|                                  |
        |                     |(6) CCR (Termination, Used-Units, |
        |                     |    QoS-Resources(QoS-Delivered)) |
        |                     |--------------------------------->|
        |                     |(7) CCA                           |
        |                     |<---------------------------------|
```

Figure 6: Example of a Diameter Credit Control with QoS Information

7.6. Classifier Examples

Example: Classify all packets from hosts on subnet 192.0.2.0/24 to ports 80, 8090 or 443 on web servers 192.0.2.123, 192.0.2.124, 192.0.2.125.

```
Classifier = {
    Classifier-Id = "web_svr_example";
    Protocol = TCP;
    Direction = OUT;
    From-Spec = {
        IP-Address-Mask = {
            IP-Address = 192.0.2.0;
            IP-Bit-Mask-Width = 24;
        }
    }
    To-Spec = {
        IP-Address = 192.0.2.123;
        IP-Address = 192.0.2.124;
        IP-Address = 192.0.2.125;
        Port = 80;
        Port = 8080;
        Port = 443;
    }
}
```

Example: Any SIP signaling traffic from a device with a MAC address of 01:23:45:67:89:ab to servers with IP addresses in the range 192.0.2.90 to 192.0.2.190.

```
Classifier = {
    Classifier-Id = "web_svr_example";
    Protocol = UDP;
    Direction = OUT;
    From-Spec = {
        MAC-Address = 01:23:45:67:89:ab;
    }
    To-Spec = {
        IP-Address-Range = {
            IP-Address-Start = 192.0.2.90;
            IP-Address-End = 192.0.2.190;
        }
        Port = 5060;
        Port = 3478;
        Port-Range = {
            Port-Start = 16348;
            Port-End = 32768;
        }
    }
}
```

7.7. QoS Parameter Examples

The following high-level description aims to illustrate the interworking between the Diameter QoS AVPs defined in this document and the QoS parameters defined in [RFC5624].

Consider the following example where a rule should be installed that limits traffic to 1 Mbit/s and where out-of-profile traffic shall be dropped. The Classifiers are ignored in this example.

This would require the Treatment-Action AVP to be set to ’shape’ and the QoS-Parameters AVP carries the Bandwidth AVP indicating the 1 Mbit/s limit. The Treatment-Action carried inside the Excess-Treatment AVP would be set to ’drop’.

In a second, more complex scenario, we consider traffic marking with Diffserv. In-profile traffic (of 5 Mbit/s in our example) shall be associated with a particular PHB-Class "X". Out-of-profile traffic shall belong to a different PHB-Class, in our example "Y".

This configuration would require the Treatment-Action AVP to be set to ’mark’. The QoS-Parameters AVPs for the traffic conforming of the profile contains two AVPs, namely, the TMOD-1 AVP and the PHB-Class AVP. The TMOD-1 AVP describes the traffic characteristics, namely, 5 Mbit/s, and the PHB-Class AVP is set to class "X". Then, the Excess-Treatment AVP has to be included with the Treatment-Action AVP set to ’mark’ and the QoS-Parameters AVP to carry another PHB-Class AVP indicating PHB-Class AVP setting to class "Y".

8. Acknowledgments

We would like to thank Victor Fajardo, Tseno Tsenov, Robert Hancock, Jukka Manner, Cornelia Kappler, Xiaoming Fu, Frank Alfano, Tolga Asveren, Mike Montemurro, Glen Zorn, Avri Doria, Dong Sun, Tina Tsou, Pete McCann, Georgios Karagiannis, Elwyn Davies, Max Riegel, Yong Li, and Eric Gray for their comments. We thank Victor Fajardo for his job as PROTO document shepherd. Finally, we would like to thank Lars Eggert, Magnus Westerlund, Adrian Farrel, Lisa Dusseault, Ralph Droms, and Eric Gray for their feedback during the IESG review phase.

9. Contributors

Max Riegel contributed the VLAN sections.

10. IANA Considerations

10.1. AVP Codes

IANA has allocated codes from the "AVP Codes" registry under Authentication, Authorization, and Accounting (AAA) Parameters for the following AVPs that are defined in this document.

Attribute Name	AVP Code	Section Defined	Data Type
QoS-Resources	508	3.1	Grouped
Filter-Rule	509	3.2	Grouped
Filter-Rule-Precedence	510	3.3	Unsigned32
Classifier	511	4.1.1	Grouped
Classifier-ID	512	4.1.2	OctetString
Protocol	513	4.1.3	Enumerated
Direction	514	4.1.4	Enumerated
From-Spec	515	4.1.5	Grouped
To-Spec	516	4.1.6	Grouped
Negated	517	4.1.7.1	Enumerated
IP-Address	518	4.1.7.2	Address
IP-Address-Range	519	4.1.7.3	Grouped
IP-Address-Start	520	4.1.7.4	Address
IP-Address-End	521	4.1.7.5	Address
IP-Address-Mask	522	4.1.7.6	Grouped
IP-Mask-Bit-Mask-Width	523	4.1.7.7	Unsigned32
MAC-Address	524	4.1.7.8	OctetString
MAC-Address-Mask	525	4.1.7.9	Grouped
MAC-Address-Mask-Pattern	526	4.1.7.10	OctetString
EUI64-Address	527	4.1.7.11	OctetString
EUI64-Address-Mask	528	4.1.7.12	Grouped
EUI64-Address-Mask-Pattern	529	4.1.7.13	OctetString
Port	530	4.1.7.14	Integer32
Port-Range	531	4.1.7.15	Grouped
Port-Start	532	4.1.7.16	Integer32
Port-End	533	4.1.7.17	Integer32
Use-Assigned-Address	534	4.1.7.18	Enumerated
Diffserv-Code-Point	535	4.1.8.1	Enumerated
Fragmentation-Flag	536	4.1.8.2	Enumerated
IP-Option	537	4.1.8.3	Grouped
IP-Option-Type	538	4.1.8.4	Enumerated
IP-Option-Value	539	4.1.8.5	OctetString
TCP-Option	540	4.1.8.6	Grouped
TCP-Option-Type	541	4.1.8.7	Enumerated
TCP-Option-Value	542	4.1.8.8	OctetString
TCP-Flags	543	4.1.8.9	Grouped

TCP-Flag-Type	544	4.1.8.10	Unsigned32
ICMP-Type	545	4.1.8.11	Grouped
ICMP-Type-Number	546	4.1.8.12	Enumerated
ICMP-Code	547	4.1.8.13	Enumerated
ETH-Option	548	4.1.8.14	Grouped
ETH-Proto-Type	549	4.1.8.15	Grouped
ETH-Ether-Type	550	4.1.8.16	OctetString
ETH-SAP	551	4.1.8.17	OctetString
VLAN-ID-Range	552	4.1.8.18	Grouped
S-VID-Start	553	4.1.8.19	Unsigned32
S-VID-End	554	4.1.8.20	Unsigned32
C-VID-Start	555	4.1.8.21	Unsigned32
C-VID-End	556	4.1.8.22	Unsigned32
User-Priority-Range	557	4.1.8.23	Grouped
Low-User-Priority	558	4.1.8.24	Unsigned32
High-User-Priority	559	4.1.8.25	Unsigned32
Time-Of-Day-Condition	560	4.2.1	Grouped
Time-Of-Day-Start	561	4.2.2	Unsigned32
Time-Of-Day-End	562	4.2.3	Unsigned32
Day-Of-Week-Mask	563	4.2.4	Unsigned32
Day-Of-Month-Mask	564	4.2.5	Unsigned32
Month-Of-Year-Mask	565	4.2.6	Unsigned32
Absolute-Start-Time	566	4.2.7	Time
Absolute-Start-Fractional-Seconds	567	4.2.8	Unsigned32
Absolute-End-Time	568	4.2.9	Time
Absolute-End-Fractional-Seconds	569	4.2.10	Unsigned32
Timezone-Flag	570	4.2.11	Enumerated
Timezone-Offset	571	4.2.12	Integer32
Treatment-Action	572	5.1	Grouped
QoS-Profile-Id	573	5.2	Unsigned32
QoS-Profile-Template	574	5.3	Grouped
QoS-Semantics	575	5.4	Enumerated
QoS-Parameters	576	5.5	Grouped
Excess-Treatment	577	5.6	Grouped
QoS-Capability	578	6	Grouped

10.2. QoS-Semantics IANA Registry

IANA has allocated a new registry under Authentication, Authorization, and Accounting (AAA) Parameters for the QoS-Semantics AVP. The following values are allocated by this specification:

(0): QoS-Desired
(1): QoS-Available
(2): QoS-Delivered
(3): Minimum-QoS
(4): QoS-Authorized

The definition of new values is subject to the Specification Required policy [RFC5226].

10.3. Action

IANA has allocated a new registry under Authentication, Authorization, and Accounting (AAA) Parameters for the Treatment-Action AVP. The following values are allocated by this specification:

0: drop
1: shape
2: mark
3: permit

The definition of new values is subject to the Specification Required policy [RFC5226].

11. Security Considerations

This document describes the extension of Diameter for conveying Quality of Service information. The security considerations of the Diameter protocol itself have been discussed in RFC 3588 [RFC3588]. Use of the AVPs defined in this document MUST take into consideration the security issues and requirements of the Diameter base protocol.

12. References

12.1. Normative References

[IEEE802.1D]	IEEE, "IEEE Standard for Local and metropolitan area networks, Media Access Control (MAC) Bridges", 2004.
[IEEE802.1Q]	IEEE, "IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks", 2005.
[IEEE802.1ad]	IEEE, "IEEE Standard for Local and metropolitan area networks, Virtual Bridged Local Area Networks, Amendment 4: Provider Bridges", 2005.
[IEEE802.2]	IEEE, "IEEE Standard for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements, Part 2: Logical Link Control", 1998.
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[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.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

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

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

12.2. Informative References

[DIAMETER-QOS] Sun, D., Ed., McCann, P., Tschofenig, H., Tsou, T., Doria, A., and G. Zorn, Ed., "Diameter Quality of Service Application", Work in Progress, October 2009.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

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

[RFC5624] Korhonen, J., Tschofenig, H., and E. Davies, "Quality of Service Parameters for Usage with Diameter", RFC 5624, August 2009.

Appendix A. MAC and EUI64 Address Mask Usage Considerations

The MAC and EUI64 address bit masks are generally used in classifying devices according to Organizationally Unique Identifier (OUI) and/or address blocks specific to the OUI assignee. The bit masks are not intended to introduce a structure into the MAC or EUI64 address space that was not intended by the IEEE.

The MAC address bit mask should be defined as a contiguous series of "N" set bits followed by a contiguous series of "48 - N" clear bits, e.g., the MAC address bit mask of 0xFF00FF000000 would not be valid. Similarly, the EUI64 address bit mask should be defined as a contiguous series of "N" set bits followed by a contiguous series of "64 - N" clear bits.

It should also be noted that some OUIs are assigned for use in applications that require number space management at the organization level (e.g., LLC/SNAP encoding), and are not commonly used for MAC addresses.

Authors' Addresses

Jouni Korhonen
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

EMail: jouni.korhonen@nsn.com

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

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

Mayutan Arumaithurai
University of Goettingen

EMail: mayutan.arumaithurai@gmail.com

Mark Jones (editor)
Bridgewater Systems
303 Terry Fox Drive, Suite 500
Ottawa, Ontario K2K 3J1
Canada

Phone: +1 613-591-6655
EMail: mark.jones@bridgewatersystems.com

Avi Lior
Bridgewater Systems
303 Terry Fox Drive, Suite 500
Ottawa, Ontario K2K 3J1
Canada

Phone: +1 613-591-6655
EMail: avi@bridgewatersystems.com