WG: Authentication, Authorization and Accounting (AAA)
Internet Draft

Gino Carrozzo
Nicola Ciulli
Gianluca Insolvibile
Giacomo Sergio
CPR

Document: draft-carrozzo-aaa-cops-maid-00.txt

Expires: November 2005   May 2005

COPS-MAID: COPS Usage for Multi-Access Inter-Domain Service Provider Networks based on MPLS-Diffserv

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [1].

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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. "

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This specification defines some extensions to the COPS protocol under the COPS-MAID client type to be used for the SP Control Plane

procedures in a Multimedia Telephony Service (MTS) context. The reference SP network infrastructure is assumed to be based on MPLS/Diffserv Control and Data Planes. The proposed architecture is defined signalling independent because it is able to uphold different signalling protocols used by the MTS-enabled terminals (e.g. SIP, H.323, H.248-Megaco, MGCP, etc.) with a common approach. The overall specification relies on the identification of a common and generalized semantic for the service requests (policy & CAC), which will encapsulate the client-specific information in a common format besides of the specific protocol (e.g. SIP, H.323, etc.).
By means of this generalized architecture it is possible to reduce the complexity of the SP Control Plane architecture.

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 [2].

Table of Contents

1. Introduction

Multimedia Telephony Services (e.g. voice/video calls, SMS/MMS services, multi-party audio/video conference) based on all-IP networks poses new challenging requirements for the infrastructures of Service Providers (SP) in terms of end-to- end Quality of Service (QoS), dynamical and tailored network resource management, security/privacy, service availability, service resiliency, etc.
All these issues are supported by the traditional PSTN for the basic telephony services, but they cannot be effectively and natively provided by the standard IP networks, which are based on the flat "best-effort" and connectionless paradigm.
Focusing on the SP network, the presence of heterogeneous multimedia applications requiring QoS guarantees on the boundary of its network entails general requirements for:

- flexibility/modularity of the SP architecture, in order to provide new multimedia services with the least possible impact on the SP internal infrastructure;
- scalability of the SP network, in order to scale well with the increasing of service demands from the users through the related access networks;
- dynamicity of the service provisioning, in order to offer dynamically (i.e. as long as the end-user needs) a transport service for the MTS with QoS guarantees (both at Layer 3 and at Layer 2);
- tailoring of the SP QoS-enabled transport services, in order to allocate the amount of network resources that exactly fulfil the end-user requirements in terms of bandwidth, delays, jitter, etc.

This specification aims to define some extensions to the COPS protocol under the COPS-MAID client type to be used in a signalling independent architecture for the SP Control Plane. This architecture satisfies the requirements for flexibility and scalability of the network, as well as dynamicity and tailoring of the related transport services. This control plane architecture is defined signalling independent because it is able to uphold different signalling protocols used by the MTS-enabled terminals (e.g. SIP, H.323, H.248-Megaco, MGCP, etc.) with a common approach, according to a "classical" User-Network Interface concept.

2. MAID network architecture

The requirement for end-to-end QoS guarantees for MTS sessions implies that the SP network must comprise QoS-capable equipment. The most effective and common choices that currently gather scalability and manageability are Diffserv [5], MPLS [6] and a mixed MPLS/ Diffserv solution [3], which merges the Control Plane features of MPLS with the differentiation and scheduling tools provided by Diffserv.
The Multi-Access Inter-Domain (MAID) SP network derives from this rationale and it relies on a mixed MPLS/Diffserv architecture (both Control and Data Planes).
The MAID Data Plane provides the mapping and the forwarding of the IP flows from the access network into the proper Diffserv PHBs/LSPs (labels) and vice versa.
The MAID Control Plane is responsible for Admission Control (AC) and policy decisions (taken on a per-flow or per-PHB basis) and for the service level agreement (SLA) maintenance.

Focusing on the Control Plane side, the SP network is a collection of functionalities and components, in support of signalling, security, billing, inter-working of domains, etc. relying on the coordination function of an MTS Call Agent (e.g. the H.323 Gatekeeper, or the SIP Proxy or a softswitch in general). After a successful phase of subscription and registration of end-users to the SP services, upon receiving a service request for an MTS call from an end-user, the SP control plane is triggered to provide:

1. the authorization, authentication and accounting of the end-user;
2. the call control through a MTS Call Agent;
3. the inter-working with peering domains of the same or different technologies (e.g. neighbouring TDM domains or other SIP/H.323 domains);
4. the setup/selection/update/release of the QoS-enabled IP service in the SP data plane that can accommodate the final MTS call, in case across a chain of SP domains.

The last step, i.e. the selection of a just running QoS-enabled IP service and/or the set-up of a new one inside the SP network, is the main QoS-related step and it relies on the action of a Transport Network Resource Manager (TNRM), also known as Bandwidth Broker (BB).

The TNRM has the main role of managing network resources by means of proper configurations of the Traffic Control (TC) modules in the involved QoS-capable IP/MPLS routers, as well as the Layer 2 priority mechanisms of the involved L2 devices. TC is the set of policing, filtering, marking, shaping and scheduling procedures available on the IP routers for QoS enforcement purposes.

The triggers for TNRM actions can be:

- requests for call admission control originated by the MTS Call Agent and related to the MTS signalling
- some provisioning information (e.g. traffic matrix like)

These two solutions are generally classified as outsourcing and provisioning approaches. Outsourcing allows a dynamic distribution of available resources, even if it requires the continuous TNRM invocation and, thus, it implies scalability issues. On the contrary, provisioning relies on the pre-allocation of a set of policies (and a certain amount of resources) by the TNRM to external Policy Enforcement Points. In this case TNRM acts as a Policy Decision Point. It would be desirable to adopt a mixed provisioning/ outsourcing scheme, in order to address both the scalability and resource optimisation issues.

The scalability issues raised by the different MTS architectures (e.g. H.323, SIP, etc.) interoperating with the single TNRM encourage the definition of a common syntax for admission control and policy requests despite of the native MTS signalling protocol (e.g. SIP, H.323, H.248, etc.) and of the SP internal network architecture (e.g. Diffserv, MPLS or both). This approach allows shifting the complexity of the network on a single module, the Signalling Translation Function (STF), which is related to the MTS Call Agent and, thus, can be distributed in different network elements.

The main actions provided by STF are:

- to map the QoS semantic of the specific MTS signalling protocol into a generalized and common semantic for the QoS-enabled IP service request;
- to provide the inter-working with the native signalling protocol dynamics (i.e. message flows), depending on the result of the admission control phase by the TNRM.

The STF translation do not extend nor modify the native MTS signalling, but it simply supports the MTS signalling by providing the missing correlation between the signalling/media negotiation and the relative actions in the SP transport network in terms of QoS configuration and enforcement.

3. The COPS-MAID rationale

This specification focuses on the possibility of integrating different MTS protocol architectures through the STF module. Since a per-protocol QoS semantic can be identified from the MTS call signalling messages, it is possible to derive a minimum set of information the TNRM must know in order to provide a correct policy decision and call admission control.

This common semantic is comprised of:

```
  - Traffic origin:
      > host which originated the flow;
      > Border Router (BR) interface where the traffic will pass
        through;
  - Traffic direction:
      > egress BR for the flow;
      > core routers the flow will go across (got from a Path
        Computation System);
  - Traffic type:
      > traffic characterization;
      > QoS parameters;
  - Temporal information (optional)
      > start time, end time, repetition intervals;
```

Besides of the specific signalling protocol used by the clients, the QoS-sensible information highlighted above MUST be passed from the STF to the TNRM via a specific protocol. Different protocols can be extended for this purpose (e.g. DIAMETER, SIP, etc.). In this specification some extensions to the COPS protocol are proposed under the COPS-MAID client type (MAID, Multiple-Access Inter-Domain).

COPS (Common Open Policy Service) [3] is a simple query and response protocol that can be used to exchange policy information in an administrative domain. Relying on the client-server model, COPS architecture is based on two fundamental elements: a policy server, called Policy Decision Point (PDP), also addressed as COPS server, and one or more policy clients, called Policy Enforcement Points (PEPs), addressed as COPS clients. At least one policy server must exist in each administrative domain, in order to implement a complete COPS communication with one or more PEPs.
A single PEP is able to support multiple client-types, and it may

send multiple Client-Open messages to the PDP, each specifying a

particular client-type to a PDP over one or more TCP connections. If a client-type is not supported by the PDP, the PDP itself can redirect the PEP to an alternative PDP address and port for a given client-type via COPS.
A client type is already defined for the integration of RSVP and COPS [4], where RSVP is COPS client-type 1. The proposed standard has some weak points due to the nature of RSVP itself, e.g. the duplication of the states installed both in PDP and in PEP which limits the scalability of the model and the possibility of expanding the core domain.
A possible solution to this scalability problem is an administrative domain with many PDP supporting one or few client-types each. Whenever a PEP sends a client-open message to a PDP, that does not support that client-type, it has the capability to redirect the PEP to the right PDP. This solution is more scalable but needs to implement many COPS servers, one for each of the supported client-types within the domain. All these COPS servers have to exchange management information to perform a coherent resource allocation, or they must query for a higher level "omniscient" Bandwidth Broker.

In this specification a unified COPS semantic is proposed in order to integrate all the different protocols supported by the access domain. This semantic will encapsulate the client-specific information in a common format besides of the specific protocol.
By means of this generalized architecture it is possible to reduce the complexity of the system, because:

- a unified COPS client-type transmits all the information to the PDP;
- it is no longer needed to develop a different COPS server (PDP) for each supported COPS client;
- there is no need to refer to a higher-level device when performing resource allocation because the unique COPS server (PDP) could be located inside the Bandwidth Broker or TNRM.

The COPS-MAID client type and the related objects are defined in the following sections.

4. COPS-MAID extensions

The meaning and usage of several COPS objects is affected when used with the COPS-MAID client type. This section describes these objects and their usage.

4.1 Client Type

COPS-MAID is COPS client-type TBD.

4.2 Context Object

The semantics of the Context object for COPS-MAID is as follows:

- R-Type (Request Type Flag)
  > R-Type = 0x01: Incoming-Message/Admission Control Requests;
  > R-Type = 0x02: Resource Allocation Request;
  > R-Type = 0x08: Configuration request;

For the COPS-MAID client-type the R-Type flag 0x04 (Outgoing-Message Request) is not used.

- M-Type (Message Type)
  > M-Type = 0x01: Add request;
  > M-Type = 0x02: Release request;
  > M-Type = 0x03: Modify request;

4.3 COPS-MAID Protocol Objects Format

All the objects described in this section must be intended as objects/attributes encapsulated within other "higher level" COPS objects. In particular, they are carried in the COPS-MAID Client Specific Information Object (ref. 5.1 for the PEP -> PDP communication) and in the Client Specific Decision Object (ref. 5.2 for the PDP -> PEP direction).
These COPS-MAID objects have a TLV format (Type-Length-Value) where Type (16 bit) identifies univocally the object, Length (16 bit) indicates the length of the object in Bytes (including the header) and Value is the content of the object.

```
    0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Type              |               Length          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             Value                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.4 Request ID Object

This object is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP, or in the Client Specific Decision data sent from a PDP to a PEP. It allows binding requests sent by the PEP and having the same COPS Handle Object with responses coming from the PDP (ref. [3]). This mechanism allows a COPS-MAID PEP to make more than one request for a specific state (identified by the Handle Object) before receiving a PDP response.

The Request ID Object is chosen by the PEP and it is opaque to the PDP. For each request a different Request ID is chosen by the PEP. Request ID values can be reused if they are associated to different Handle Objects.

```
 0             7              15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 1           |         Length = 8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Request ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.5 Source Host IPv4 address Object

This object specifies the IPv4 address of the host originating the flow for which a PDP decision is requested. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP. The host IP address could be useful to the TNRM in order to perform Authorization operations.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 2           |         Length = 8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Source Host IPv4 Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.6 Destination IPv4 address Object

This object specifies the IPv4 address of the host which is the destination of the flow that requires a PDP decision. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.
The destination address is needed each time the MTS Call Agent is not aware of the egress router for the traffic flow for which is performing the request. Moreover, the destination address could be useful to the TNRM in order to perform flow authorization operations.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 3           |         Length = 8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Destination Host IPv4 Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.7 Source Host IPv6 address Object

This object specifies the IPv6 address of the host originating the flow for which a PDP decision is requested. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 4            |         Length = 20           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                |
|                     Source Host IPv6 Address                   |
|                                                                |
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.8 Destination IPv6 address Object

This object specifies the IPv6 address of the host, which is the destination of the flow that requires a PDP decision. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.
The destination address is needed each time the MTS Call Agent is not aware of the egress router for the traffic flow for which is performing the request.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 5            |         Length = 20           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                |
|                 Destination Host IPv6 Address                  |
|                                                                |
|                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.9 IPv4 Ingress Border Router Interface Object/TrunkID

This object specifies the IPv4 address of the ingress BR output interface for the traffic flow which requires resource allocation. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 6           |         Length = 16           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Ingress BR Output IPv4 Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Label Type                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Label                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Label Type field is used to specify the semantics of the following label object:

- Label Type = 0x00000001: Diffserv;
- Label Type = 0x00000002: MPLS;
- Label Type = 0x00000003: ATM;
- Label Type = TBA;

The Label object specifies an identifier for the traffic flow with respect to the specified semantic type (e.g. a DSCP in case of Diffserv semantics, a label in case of MPLS, etc.). The couple IPv4 output address/label identifies univocally the trunk where the traffic flow has to be aggregated.

4.10 IPv6 Ingress Border Router Interface Object/TrunkID

This object specifies the IPv6 address of the ingress border router output interface for the traffic flow which requires resource allocation. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 7           |         Length = 28           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|               Ingress BR Output IPv6 Address                  |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Label Type                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Label                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Label Type field is used to specify the semantics of the label:

- Label Type = 0x00000001: Diffserv;
- Label Type = 0x00000002: MPLS;
- Label Type = 0x00000003: ATM;
- Label Type = TBA

The Label value is set according to the specified Label Type.

4.11 Egress Border Router Interface Object

This object specifies the IPv4 Egress Border Router interface address for the traffic flow for which the resource allocation request is performed. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 8           |         Length = 8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Egress BR Input IPv4 Address                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.12 IPv6 Egress Border Router Interface Object

This object specifies the IPv6 Egress Border Router interface address for the traffic flow for which the resource allocation request is performed. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 9           |         Length = 20           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                  Egress BR Input IPv6 Address                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.13 Traffic Type Object

Traffic Type Objects are maintained in the form of a list in which each Traffic Type ID value corresponds to a predefined (either standardized or client defined) characterization:
Traffic Type ID = value -> characterization.

- Traffic Type ID = TBA
- Traffic Type ID = TBA

Traffic Type Object and Traffic Characterization Object, described in the next paragraph, can be used together or separately, allowing to specify traffic parameters at different levels.

```
   0             7               15                             31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Type = 10          |        Length = 16            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Traffic Type ID                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.14 Traffic Characterization Object

The traffic characterization object describes the traffic characterization parameters for the traffic flow. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP.

```
   0             7               15                             31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Type = 11          |        Length = var           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Exclude-any                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Include-any                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Include-all                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Setup Prio  | Holding Prio  |        MPLS/DS TSpec number   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        MPLS/DS Tspec  sub-objects             |
   |                              ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Resource affinities are declared by three 32 bit masks:

- Exclude-any (32-bit), for the set of attribute filters associated with a tunnel any of which renders a link unacceptable;
- Include-any (32-bit), for the set of attribute filters associated with a tunnel any of which renders a link acceptable)
- Include-all (32-bit), for the set of attribute filters associated with a tunnel all of which must be present for a link to be acceptable.

Priorities for the flow are specified by:

- Setup Priority (in the range of 0 to 7), which specifies priority of the session with respect to taking resources;
- Holding Priority (in the range of 0 to 7), which specifies priority of the session with respect to holding resources;

The MPLS/DS TSpec number specifies the total number of MPLS/DS Traffic Specification entries (sub-objects) contained in the object. Each sub-object is structured as follows:

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |       DS Behaviour Class      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Traffic Profile Type                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Profile Characterization                 |
|                              ...            |
|                              ...            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Types for MPLS/DS Tspec sub-object are:

- Type = 0x0001: E-LSP Tspec;
- Type = 0x0002: L-LSP Tspec;

and depending on Type value the DS Behaviour Class field contains the EXP value (3 bit û left aligned) or the PHBID field (16 bit, ref.[8]).

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type = 1         | EXP |         Reserved        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...            |
```

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type = 2         |             PHBID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...            |
```

The Traffic Profile Type value specifies the type of characterization attached to the object.
The following characterization types are supported:

- Traffic Profile Type 1 = 0x00000001: Average & Peak Rate (r, p, m, M);
- Traffic Profile Type 2 = 0x00000002: LBAP point (r, b, p, m, M,

```
                                            avgr);
     - Traffic Profile Type 3 = 0x00000003: 3D LBAP point (l, r, b, p,
                                            M, avgr);
     - Traffic Profile Type 4 = 0x00000004: LBAP plot (p, m, M, avgr,
                                            (r, b), ... , (r, b));
     - Traffic Profile Type 5 = 0x00000005: 3D-LBAP plot (p, m, M,
                                            avgr, (l, (r,b) plot,...,
                                            l, (r,b)plot));
```

where r indicates the rate (bytes/s), b the bucket size (bytes), p the peak rate (bytes/s), m the minimum policed unit (bytes), M the connection MTU (bytes), avgr the average rate (bytes/sec) and l the loss probability.

4.14.1 Profile Characterization for Type 1

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              Rate                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Peak Rate                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Minimum policed unit                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Connection MTU                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.14.2 Profile Characterization for Type 2

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Rate                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Bucket Size                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Peak Rate                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Minimum policed unit                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Connection MTU                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Average Rate                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.14.3 Profile Characterization for Type 3

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Loss Probability                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Rate                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Bucket Size                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Peak Rate                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Minimum policed unit                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Connection MTU                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Average Rate                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.14.4 Profile Characterization for Type 4

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Peak Rate                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Minimum policed unit                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Connection MTU                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Average Rate                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Rate                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Bucket Size                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Rate                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Bucket Size                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.14.5 Profile Characterization for Type 5

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             Peak Rate                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Minimum policed unit                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Connection MTU                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Average Rate                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Loss Probability                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Rate                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Bucket Size                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Rate                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Bucket Size                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              ...            |
   |                              ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Loss Probability                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Rate                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Bucket Size                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Rate                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                            Bucket Size                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.15 QoS Class Description Object

This object specifies the QoS class to which the traffic flow belongs to. Four different QoS classes are defined:

- Class Type = 0x00000001: bounded losses;
- Class Type = 0x00000002: guaranteed bandwidth;
- Class Type = 0x00000003: guaranteed bandwidth + bounded delay;
- Class Type = 0x00000004: guaranteed bandwidth + bounded delay + bounded jitter.

Each one of these classes has more stringent QoS guarantees than the previous class.

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Type = 12          |         Length = 8            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Class Type                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.16 QoS Parameters Description Object

This object defines the QoS parameters requested. Four different parameters are defined:

- bandwidth;
- delay;
- jitter;
- loss probability;

For each of the previous parameters a slack term could be indicated: while the bandwidth slack term must be subtracted to the bandwidth parameter in order to find the minimum allowed bandwidth, the jitter and delay slack terms must be added to the respective parameters in order to find the maximum allowed delay and jitter.

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Type = 13          |         Length = 28           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Bandwidth (bytes/sec)                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                      Bandwidth Slack term                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Jitter (msec)                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Jitter Slack term                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Delay  (msec)                         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Delay Slack term                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.17 LSP Recovery description Object

This object is used by the PEP to request by the way of a PDP a Path Computation System (PCS) to compute a pair of Explicit Route objects for a traffic flow. It is carried in the Client Specific Resource

Allocation Request data Object of a Decision Message sent from a PDP to a PEP.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 14          |         Length = 8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Recovery type          |        Recovery diversity     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Values defined for recovery type are:

- Recovery type = 0x0000: Unprotected
- Recovery type = 0x0001: Path Protection
- Recovery type = 0x0002: Link Protection
- Recovery type = 0x0003: Path Fast-Restoration (with preplanning of a backup ERO)
- Recovery type = 0x0004: Link Fast-Restoration (with preplanning of a partial detour for each link)
- Recovery type = TBA

Values defined for recovery diversity are:

- Recovery diversity = 0x0000: None
- Recovery diversity = 0x0001: Node
- Recovery diversity = 0x0002: Link
- Recovery diversity = 0x0003: SRLG
- Recovery diversity = TBA

4.18 COPS-MAID Decision Object

This object is used by the PDP to inform the PEP how to aggregate the flow. It is carried in the Client Specific Decision Data Object of a Decision Message sent from a PDP to a PEP.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 15          |         Length = 12           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Label Type                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Label                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Label Type field is used to specify the semantics of the label:

- Label Type = 0x00000001: Diffserv;
- Label Type = 0x00000002: MPLS;
- Label Type = 0x00000003: ATM;
- Label Type = TBA;

The Label value is set according to the specified Label Type.

4.19 Temporal information object

The temporal information object is used to specify the time interval inside which the request is valid. It is carried in the Client Specific Information Object of a Request Message sent from a PEP to a PDP or in the Client Specific Decision data sent from a PDP to a PEP. y means of this object one or more time intervals and/or periodically repeated time intervals could be defined.
An ASCII string is used to describe the temporal information. If the string length is not multiple of 32 bits appropriate zero padding bytes are used.

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Type =16           |         Length = var          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           ASCII String                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              ...            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The ASCII string is in the form:

```
    {[<start date _ end date>
      <start day of the week _ end day of the week>
      <start hour _ end hour>]
     [<start date _ end date>
      <start day of the week _ end day of the week>
      <start hour _ end hour>]
    ...
     [<start date _ end date>
      <start day of the week _ end day of the week>
      <start hour _ end hour>]
    }
```

where:

- start date and end date are in the format: "ddmmyyyy";
- start day of the week and end day of the week are expressed by one of the following character: A-B-C-D-E-F-G. The letters are the coding for: sun - mon - tue - wen - thu - fri - sat , respectively;
- start hour and end hour are in the format: "hhmm".

Blanks are used for increase the readability and are suppressed in the TempInfo object value.

The ASCII string starts with "{" and ends with "}". Each time interval is specified internally to the characters "[" and "]". This time interval is the intersection of the time intervals (separated by the character "-") specified internally to the characters "<" and ">". Otherwise the global time interval is the union of all the time intervals specified internally to the characters "[" and "]". The character ("*") is used as wildcard.

As an example, the ASCII string

```
    {[<15092001-15122001><B-D-F><1600-1800>]
     [<01042002-30062002><*><1700-1900>]}
```

means: every Monday, Wednesday and Friday between September the 15th and December the 15th 2001, from 16:00 to 18:00 and everyday between April the 1st and June the 30th 2002, from 17:00 to 19:00.

4.20 **Explicit Route object**

The contents of an Explicit Route object are a series of variable-length data items called sub-objects. An explicit route is a particular path in the network topology. An explicit route is described as a list of groups of nodes along the explicit route. In addition to the ability to identify specific nodes along the path, an explicit route can identify a group of nodes (called abstract node) that must be traversed along the path. This capability allows the routing system a significant amount of local flexibility in fulfilling a request for an explicit route. This capability allows the generator of the explicit route to have imperfect information about the details of the path.
The explicit route is encoded as a series of sub-objects contained in an ERO object. Each sub-object identifies a group of nodes in the explicit route. An explicit route is thus a specification of groups of nodes to be traversed.
This object is used by the PDP to inform the MPLS-enabled ingress BR (LER) of the MPLS/Diffserv SP network that an MPLS signalling session (e.g. via RSVP-TE protocol) must be initiated to setup an LSP. It is carried in the Client Specific Decision Data Object of a Decision Message sent from a PDP to a PEP.

```
 0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Type = 17          |         Length = var          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              ...             |
|                           Sub-objects                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.20.1 ERO Sub-objects

Each sub-object has the form:

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
   |L|    Type     | Length = var  |                                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                                +
   |                              ...            |
   |                           Sub-object contents                  |
   |                                                                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
```

The L bit is an attribute of the sub-object. The L bit is set if the sub-object represents a loose hop in the explicit route. If the bit is not set, the sub-object represents a strict hop in the explicit route. The path between a strict node and its preceding node MUST include only network nodes from the strict node and its preceding abstract node. The path between a loose node and its preceding node MAY include other network nodes that are not part of the strict node or its preceding abstract node.
The Type indicates the type of contents of the sub-object. Currently defined values are:

- Type = 0x01: IPv4 prefix
- Type = 0x02: IPv6 prefix
- Type = 0x20: Autonomous system number

The Length contains the total length of the sub-object in bytes, including the L, Type and Length fields. The Length MUST be at least 4, and MUST be a multiple of 4.

4.20.1.1 **Sub-object 1:** IPv4 prefix

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
   |L| Type = 0x01 |  Length = 8   |    Ingress i/f IPv4 address    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
   | Ingress i/f IPv4 addr (contd) | Prefix length |    Padding     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
```

The Ingress interface IPv4 address (32) is treated as a prefix based on the prefix length (8 bit) value below. Bits beyond the prefix are ignored on receipt and SHOULD be set to zero on transmission.
Note that a prefix length of 32 indicates a single IPv4 node.

4.20.1.2 **Sub-object 1:** IPv6 prefix

```
   0             7               15                              31
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+--+
```

```
|L| Type = 0x02 |  Length = 20  |    Ingress i/f IPv6 address   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Ingress i/f IPv6 addr (contd)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Ingress i/f IPv6 addr (contd)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Ingress i/f IPv6 addr (contd)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ingress i/f IPv4 addr (contd) | Prefix length |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Ingress interface IPv6 address (128) is treated as a prefix based on the prefix length (8 bit) value below. Bits beyond the prefix are ignored on receipt and SHOULD be set to zero on transmission.
Note that a prefix length of 128 indicates a single IPv6 node.

4.20.1.3 **Sub-object 32:** Autonomous System Number

The contents of an Autonomous System (AS) number sub-object are a 4-octet AS number. The abstract node represented by this sub-object is the set of nodes belonging to the autonomous system.

```
0             7               15                              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L| Type = 0x20 |  Length = 8   |          AS Number            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.21 Reject Reason Object

This object specifies the reason of a negative answer coming from the PDP. It is carried in the Client Specific Decision Data Object of a Decision Message sent from a PDP to a PEP.
Reason Code:

- Reason Code = 0x00000001: Resource unavailable;
- Reason Code = 0x00000002: Unsupported Traffic Type;
- Reason Code = 0x00000003: Unacceptable Src Address;
- Reason Code = 0x00000004: Unacceptable Dst Address;
- Reason Code = 0x00000005: Invalid Traffic Parameters;
- Reason Code = 0x00000006: No primary route to host;
- Reason Code = 0x00000007: No backup route to host;
- Reason Code = 0x00000008: No diversity for backup route;
- Reason Code = ...

5. COPS-MAID Client Specific Information Object

5.1 COPS-MAID Client Specific RAR data

The COPS-MAID Client Specific Resource Allocation Request data (RAR) is carried in the REQ messages for the COPS-MAID client and contains the description of the resources and has a different format depending on whether the type of request is Add/Release or Modify.
For Add and Release messages the COPS-MAID ClientSI RAR is:

```
    <ClientSI: COPS-MAID RAR data>::=  <ReqID>
                                       [<SrcIPv4(s)>]
                                       [<SrcIPv6(s)>]
                                       [<DstIPv4(s)>]
                                       [<DstIPv6(s)>]
                                       [<IPv4BRIf(s)>]
                                       [<IPv6BRIf(s)>]
                                       [<IPv4ER(s)>]
                                       [<IPv6ER(s)>]
                                       [<Ttype(s)>]
                                       [<TDesc(s)>]
                                       [<QoSDesc(s)>]
                                       [<QoSPrms(s)>]
                                       [<Recov(s)>]
                                       [<TempInfo(s)>]
```

with:

```
      X(s) ::= <X> | <X> <X>
```

whether for the Modify messages is:

```
<ClientSI: COPS-MAID RAR data> ::=  <ReqID>
                                    [<SrcIPv4(s)>] (new)
                                    [<SrcIPv4(s)>] (old)
                                    [<SrcIPv6(s)>](new)
                                    [<SrcIPv6(s)>] (old)
                                    [<IPv4BRIf(s)>](new)
                                    [<IPv4BRIf(s)>](old)
                                    [<IPv6BRIf(s)>](new)
                                    [<IPv6BRIf(s)>](old)
                                    [<IPv4ER(s)>](new)
                                    [<IPv4ER(s)>](old)
                                    [<IPv6ER(s)>] (new)
                                    [<IPv6ER(s)>] (old)
                                    [<Ttype(s)>] (new)
                                    [<Ttype(s)>] (old)
                                    [<TDesc(s)>] (new)
                                    [<TDesc(s)>] (old)
```

```
                                        [<QoSDesc(s)>] (new)
```

```
                                            [<QoSDesc(s)>] (old)
                                            [<QoSPrms(s)>] (new)
                                            [<QoSPrms(s)>] (old)
                                            [<Recov(s)>] (new)
                                            [<Recov(s)>] (old)
                                            [<TempInfo(s)>] (new)
                                            [<TempInfo(s)>] (old)
```

5.2 COPS-MAID Client Specific Decision data

The COPS-MAID ClientSI Decision data is carried in the COPS decision message and contains the PDP decision for a certain Request ID.

```
   <Client SI: COPS-MAID Decision data> ::=  <Request ID>
                                             < COPS-MAIDDec> | < RejRea>
                                             [<SrcIPv4(s)>]
                                             [<SrcIPv6(s)>]
                                             [<IPv4BRIf(s)>]
                                             [<IPv6BRIf(s)>]
                                             [<IPv4ER(s)>]
                                             [<IPv6ER(s)>]
                                             [<Ttype(s)>]
                                             [<TDesc(s)>]
                                             [<QoSDesc(s)>]
                                             [<QoSPrms(s)>]
                                             [<TempInfo(s)>]
                                             [<ERO(s)>] - primary
                                             [<ERO(s)>] û backup
```

6. Message content

6.1 Request Message (REQ) PEP -> PDP

```
   <Request> ::=  <Common Header>
                  <Client Handle>
                  <Context = Resource-Allocation request>
                  [<IN-Int>]
                  [<OUT-Int>]
                  [<ClientSI: COPS-MAID RAR data>]
                  [<Integrity>]
```

6.2 Decision Message (DEC) PDP -> PEP

```
   <Decision Message> ::=  <Common Header>
```

```
                           <Client Handle>
                           <Decision> | <Error>
                           [<Integrity>]

   <Decision>::=  <Context = Resource-Allocation request>
                  <Decision: Flags>
                  <Decision: COPS-MAID decision data>
```

6.3 Report State Message (RPT) PEP -> PDP

The RPT Message is sent by the PEP to PDP in case of problems with a received Decision Message.
RPT Message has the following format:

```
   <Decision Message> ::=  <Common Header>
                           <Client Handle>
                           <Report Type>
                           <Client SI: COPS-MAID RPT data>
                           [<Integrity>]

   <Client SI: COPS-MAID RPT data> ::=  <Request ID>
```

Security Considerations

The extensions proposed in this document do not raise any new security concerns with respect to those declared in [3].

IANA Considerations

Within this document a new COPS Client Type (COPS-MAID) is defined and it needs to be assigned in compliancy with the IANA assignments for the COPS protocol parameters:

http://www.iana.org/assignments/cops-parameters

A Client-type values within the range 0x0001-0x3FFF (Specification Required) could apply to this specification, since the behavior and applicability of the COPS-MAID extensions are hereby described.
The other extensions described in this document are in the form of TLV sub-objects carried by the COPS-MAID Client Specific Information Object (ref. 5.1 for the PEP -> PDP communication) and in the Client Specific Decision Object (ref. 5.2 for the PDP -> PEP direction).

The Type values assigned by authors for each of them from sec. 4.4 to sec. 4.21 are those suggested for IANA assignment.

Normative References

[1] Bradner S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[2] Bradner S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[3]Durham D., Ed., "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[4] Herzdog S., Ed., "COPS Usage for RSVP", RFC 2749, January 2000.

Informative References

[5]Blake S. et al., "An Architecture for Differentiated Services", RFC 2475, December 1998.

[6] Rosen E. et al., "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[7] Le Faucheur F., Ed., "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", RFC 3270, May 2002.

[8] Black D. et al., "Per Hop Behavior Identification Code", RFC 3140, June 2001.

Acknowledgments

This work has been supported by the European Commission through the Integrated Project "Multimedia Networking (MediaNet)" - IST-Project no. FP6-507452.

Author's Addresses

Gino Carrozzo
Divisione Informatica e Telecomunicazioni - Consorzio Pisa Ricerche
C.so Italia, 116
56100 Pisa, ITALY

Email: g.carrozzo@cpr.it

Nicola Ciulli
Divisione Informatica e Telecomunicazioni - Consorzio Pisa Ricerche
C.so Italia, 116
56100 Pisa, ITALY
Email: n.ciulli@cpr.it

Gianluca Insolvibile
Divisione Informatica e Telecomunicazioni - Consorzio Pisa Ricerche
C.so Italia, 116
56100 Pisa, ITALY
Email: g.insolvibile@cpr.it

Giacomo Sergio
Divisione Informatica e Telecomunicazioni - Consorzio Pisa Ricerche
C.so Italia, 116
56100 Pisa, ITALY
Email: g.sergio@cpr.it

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Information on the IETF's procedures with respect to rights in IETF Documents can be found in BCP 78 and BCP 79.

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

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

Disclaimer of Validity

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

Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.