Integrated services over DiffServ network using COPS-ODRA

February 22, 2000

Status of this Memo

Copyright Notice

Abstract

This document details the IntServ operation over a DiffServ network in
the case of dynamic admission control procedure supported by the COPS
protocol. An RSVP aware Edge Router uses the COPS protocol in order to
query a Bandwidth Broker, which manages the resources within the
DiffServ domain. This document relies on the COPS extensions proposed in
the companion draft "COPS usage for Outsourcing DiffServ Resource
Allocation" [COPS-ODRA].

        Table of Contents

Glossary

API   Application Programming Interface
BB   Bandwidth Broker
COPS   Common Open Policy Service
ER   Edge Router
LPDP   Local Policy Decision Point
LLDAL   Link Layer Dependent Adaptation Layer
ODRA   Outsourcing Diffserv Resource Allocation
PDP   Policy Decision Point
PEP   Policy Enforcement Point
QoS   Quality of Service
RSVP   ReSerVation Protocol
SLS   Service Level Specification

1.   Introduction

The Integrated Service Architecture (IntServ _ see [INTSERV] and
[RFC2210]) and the Differentiated Service Architecture (DiffServ _ see
[2BIT] and [DSARCH]) have been proposed in the IETF to address the
support of QoS in IP networks. A combination of the two approaches is
proposed in [INTDIF], which provides a general framework while leaving
open several architectural issues. In particular three options for
resource management in the DiffServ network are listed: 1) statically
provisioned resources, 2) dynamically provisioned resources by means of
RSVP, 3) dynamically provisioned resources by other means (e.g., a form
of Bandwidth Broker).

In this draft we consider the third scenario where the Admission Control

in the DiffServ network is based on a Bandwidth Broker. The Edge Router
communication with the Bandwidth Broker is realized by means of an
extension to the COPS protocol described in the related proposed
document [COPS-ODRA]. The COPS is used to exchange resource allocation
requests/responses. The Edge Router contains the PEP _ Policy
Enforcement Point (client side of the COPS protocol), while the

Bandwidth Broker plays the role of the PDP _ Policy Decision Point
(server side of the COPS protocol).

The goal of this document is to define the scenario for the interworking
of RSVP and COPS protocols and to describe the interaction between them.
Note that in the Edge Router, the implementations of the RSVP and of the
COPS protocols can be seen as two different entities that need to
communicate. The detailed description of the related communication
mechanism is outside the scope of this document. A proposal for such
communication mechanism, based on an API library is described in a
companion draft [CCAPI].

2. IntServ over DiffServ: the role of the Edge Router

This section briefly recall the main concepts behind the IntServ over
DiffServ operation as described in [INTDIF]; it focuses on the specific
scenario of dynamic admission control and analyzes in detail the role of
the Edge Router. The reference network configuration is depicted in
Figure 1.

```
              _____          _____         _____
            /          \      /          \     /         \
          /              \   /              \  /           \
   +---+ |                |---|              |---|          | +---+
   |Tx |-|                |ER1|              |ER2|          |-| Rx|
   +---+ |                |-- |              |---|          | +---+
          \              /   \              /  \           /
            _____/       _____/     _____/

       IntServ Access Region    DiffServ Core   IntServ Access Region
```
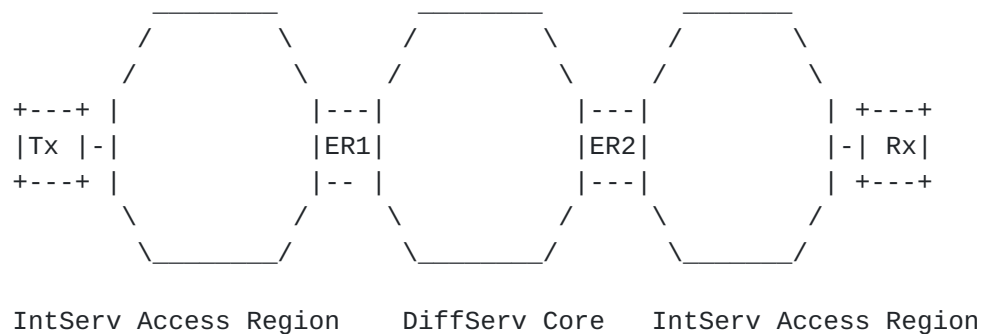
                 Figure 1: Reference Network Configuration

As explained in [INTDIF], such a solution tries to conjugate benefits of
both the IntServ and the DiffServ approaches to QoS provisioning. In
fact, it achieves scalability due to the DiffServ aggregation in the

core, while keeping the advantages of end-to-end signaling.

The Edge Router (ER) plays a key role in this scenario. The ER is the router placed at the boundary between the IntServ access network and the DiffServ core. Differently from core routers, it is RSVP aware and stores per-flow states; nevertheless, it is capable of managing packets both on a micro-flow basis and on an aggregate basis. The choice between the two possibilities depends on the role of the Edge Router. When it acts as ingress ER, i.e. for packets from the originating IntServ network to the DiffServ core it forwards packets in an aggregate fashion on the outgoing interface, while it can handle micro-flows on the incoming interface. Instead, when it behaves as egress ER, i.e. for packets from the DiffServ core to the destination IntServ network, it is

able to distinguish micro-flows on the outgoing interface. Note, however, that the distinction between ingress and egress edge router depends only on the direction of the data stream. This means that the same ER may be ingress ER for a flow and egress ER for another one in the opposite direction.

As previously stated, the Edge Router is one of the main actors in the situation depicted in Figure 1. This is especially true for the ingress Edge Router, since it provides some important functionality. Among them we cite:

 - Classification: it performs per micro-flow classification, i.e. it is
       able to distinguish the different Intserv flows.
 - Mapping: it performs mapping of IntServ service classes into DiffServ
       PHBs; it is also in charge of aggregating IntServ micro-flows
       into DiffServ aggregates.
 - Marking: it marks (or remarks) the DS field of incoming packets
       according to the target PHB, that in turn results from the
       mapping operation explained above.
 - ADSPEC update: for GS flows the exported terms in the ADSPEC (i.e. C
       and D) must be properly updated with a value depending upon the
       topology and the characteristics of the DiffServ core.
 - Admission Control: this is one of the main tasks performed by the
       ingress Edge Router. The Admission Control is applied to the
       virtual hop represented by the DiffServ core network. The
       purpose of this procedure is to ensure that Diffserv resources
       are available in the Diffserv domain to support the requested
       Intserv flow. In a "pure" DiffServ network per flow Admission
       Control is not needed, as simpler "aggregate" policing at
       ingress points based on provisioning can be used. As explained
       in [INTDIFF], the purpose of per-flow admission control is to
       increase network utilization and/or to support tighter end-to-
       end QoS guarantees (at the expense of increased complexity).

Let us focus on the Admission Control functionality performed by the
Edge Router at the ingress boundary of the DiffServ cloud. There are
basically two distinct approaches to this task:

- Distributed: in this case Admission Control is based on information
            locally available in the ingress Edge Router.
- Centralized: differently from the previous choice, Admission Control
            is realized by means of a Bandwidth Broker (BB), i.e.
            logically centralized entity acting as an Admission Control
            server. The BB could use global knowledge of both the
            network topology and the resource allocation (see Figure 2)
            to take Admission control decisions. The definition of
            mechanisms and algorithms used for the BB operation is
            outside the scope of this document and will not be further
            detailed here. Related work can be found in [IWQOS99].

```
                          +------+
                    +----|  BB  |----+
                    |     +------+    |
           _____  |    _____    |    _____
          /        \ |  /        \  | /        \
         /          \| /          \ | /          \
    +---+ |          |---|          |---|          | +---+
    |Tx |-|          |ER1|          |ER2|          |-| Rx|
    +---+ |          |-- |          |---|          | +---+
          \          /   \          /   \          /
           _____/     _____/     _____/

     IntServ Access Region    DiffServ Core    IntServ Access Region
```

                    Figure 2: Centralized Bandwidth Broker

The distributed solution is quite simple to implement, but it is also
rather inaccurate, since each ER exploits information with a local
scope, without the overall vision of the network status. In contrast,
the second approach raises some non-trivial issues in terms of
complexity and scalability, but allows better resource utilization
within the DiffServ cloud.

In this document the centralized solution of Figure 2 is analyzed. A
protocol for the communication between the ER and the centralized server
is needed. The use of the COPS [COPS] is considered here. The COPS

protocol can be extended to support new Client types. The extensions
proposed in [COPS-ODRA] are suited to support the Intserv operation on
Diffserv network.

Note that the scenario depicted in Figure 2 refers to a single DiffServ
core domain (intra-domain case). In a multi-domain scenario, one could
simply use of RSVP as end-to-end signaling (raising scalability
concerns) or more complex communication mechanisms between BBs of
different domains could be defined. These inter-domain aspects are
outside the scope of this document.

3. RSVP/COPS interaction

From now on the architecture represented in Figure 2 will be assumed as
the reference scenario. As previously explained, the Edge Router
comprises most of the functionality needed for interworking, including
admission control on a micro-flow basis. Obviously, for proper
operation, RSVP and COPS signaling should properly interact. To clarify
this mechanism let us consider the sequence of operations involved in an
end-to-end reservation, that is described in detail in [INTDIF] and
reported here in a simplified form. We are considering unicast
reservations:

  - The sender application on Tx generates an RSVP PATH message
    carrying the flow's characteristics (i.e. the Tspec). The message
    is carried towards the receiver Rx; in the IntServ access regions
    it is subject to standard RSVP/IntServ processing, while in the
    DiffServ core it is forwarded transparently.

  - When the receiving host Rx receives the PATH it generates the
    corresponding RSVP RESV message, that is carried back towards the
    sending host. As before, the RESV message undergoes standard
    RSVP/IntServ processing in the IntServ access regions and is simply
    ignored in the DiffServ core. During his travel backwards, assuming
    that it not rejected before for resource unavailability, it reaches
    the ingress Edge Router ER1.

  - In ER1, the RESV message triggers a query to the Bandwidth Broker.
    The latter, based on the information carried in the RESV message
    and on its view on the utilization of network resources, decides to
    admit or reject the request. The total amount of available
    resources is determined by Diffserv provisioning mechanisms.

  - If ER1 approves the request, the RESV message is admitted and is

allowed to continue upstream towards the sender. If it rejects the
request, the RESV is not forwarded and the appropriate RSVP error
messages are sent.

As described in the list above, queries from the ingress ER to the BB
are triggered whenever the former receives a RESV message from the
downstream DiffServ domain. Note that in this situation RSVP and COPS
are synchronized. A possible choice in the implementation of RSVP/COPS
interaction would be to keep this synchronization. This would imply a
blocking behavior; whenever the RSVP daemon triggers a query it blocks
indefinitely waiting for a response. This raises the obvious problem of
managing situations in which a response from the PDP/BB is not
temporarily available. In fact, in such cases, the blocking behavior
should be avoided, in order to be able to manage other events and to
avoid soft state expirations.

As a consequence the RSVP should be properly adapted in order to manage
requests and responses asynchronously. After a query and before getting
the response, processing of other events should be possible. A possible
way to realize this behavior could be the introduction of pending
reservation states in the RSVP daemon. Consider as an example the
reception of a RESV message by the ingress Edge Router ER1, that in turn
triggers an admission control query towards the PDP/BB. In this case the
RSVP daemon could set up a reservation state before retrieving the
response, marking it as pending. No further operations related to this
state should be performed until it remains pending, i.e. until the
response from the PDP/BB is available; this means, for example, that the
RESV message should not be forwarded upstream. In the case of positive
response (i.e. reservation request accepted), the corresponding
reservation should be instantiated on the egress interface of ER1, and
the RESV message should be forwarded upstream. In the case of rejection,

e.g. due to unavailability of resources, a blockade state should be set
up in ER1 and a RESVERR message should be propagated downstream towards
Rx, according to standard RSVP behavior. The introduction of pending
reservation states could allow the RSVP daemon to perform normal
processing related to states other than the one considered. This
obviously includes refresh operations triggered by timeout expirations.
Note that a timeout could also be defined for pending states. In this
case, when the timeout of a pending reservation state expires, and no
response has been received in the meanwhile, the RSVP could either
reject the request or revert to local decision based on LPDP
functionality in the ER.

Figure 4 and Figure 5 show the typical sequence of operations involved
in a reservation, both in the case of acceptance and in the case of

rejection. As stated above, the interface between the PEP in the ER and
the PDP/BB is realized by means of the COPS protocol extension described
in [COPS-ODRA], while the interface between the RSVP daemon and the PEP
within the ER is described in a companion draft [CCAPI].

```
                                  +---------+
                    +-------------| PDP/BB  |----------------+
                    |             +---------+                |
              +---------+                              +---------+
              |  ER1    |                              |  ER2    |
              +---------+                              +---------+
            RSVP      PEP          PDP/BB             RSVP


              |         |     RSVP RESV MESSAGE           |
              |<-------+---------------+-----------------|
              |         |              |                 |
              |request  |              |                 |
              |message  | COPS-ODRA    |                 |
      +       |------->| REQ MESSAGE   |                 |
      |       |         |--------------->|                 |
   pending    |         | Add          |                 |
   reserv.    |         |              |                 |
    state     |         | COPS-ODRA    |                 |
      |       |dispatch| DEC MESSAGE   |                 |
      +       |message |<--------------|                 |
      +       |<-------| Install       |                 |
   normal     |   OK   |              |                 |
  reservation|         |              |                 |
   state      |         |              |                 |
      |       |         |              |                 |
     ...     ...       ...            ...               ...
```

                  Figure 3: Reservation successfully accepted

      Integrated services over DiffServ network using COPS-ODRA Feb-00

```
                                  +---------+
                    +-------------| PDP/BB  |----------------+
                    |             +---------+                |
              +---------+                              +---------+
              |  ER1    |                              |  ER2    |
              +---------+                              +---------+
            RSVP      PEP          PDP/BB             RSVP
```

```
           |            |      RSVP RESV MESSAGE              |
           |<-------+---------------+------------------|
           |request |               |                   |
           |message | COPS-ODRA     |                   |
    +      |------->| REQ MESSAGE   |                   |
    |      |        |-------------->|                   |
 pending   |        | Add           |                   |
 reserv.   |        |               |                   |
 state     |        | COPS-ODRA     |                   |
    |      |dispatch| DEC MESSAGE   |                   |
    +      |message |<--------------|                   |
    +      |<-------| Remove        |                   |
    |      |  NO    |     RSVP RESVERR MESSAGE           |
 blockade  |--------+---------------+----------------->|
  state    |        |               |                   |
    |      ...      ...             ...                 ...
```

Figure 4: Reservation rejected

Finally, Figure 5 shows the sequence of events involved in the release
of resources. When a RESV TEAR is received a COPS-ODRA REQ message is
sent to the PDP/BB, in order to let it keep track of resource usage.
After the reception of the corresponding DEC message the PEP notifies
the RSVP, which in turn releases the reservation and forward upstream
the RESV TEAR. Note however that the same sequence of operations happens
in the case of PATH TEAR or in the case of timeout expiration.

      Integrated services over DiffServ network using COPS-ODRA Feb-00

                              +---------+
                  +---------------| PDP/BB |----------------+
                  | +---------+ |
```

```
             +---------+                              +---------+
             |  ER1    |                              |  ER2    |
             +---------+                              +---------+
          RSVP      PEP              PDP/BB         RSVP

             |         |    RSVP RESV TEAR MESSAGE       |
             |<-------+--------------+------------------|
             |request |             |                  |
             |message | COPS-ODRA   |                  |
      +      |------->| REQ MESSAGE |                  |
      |      |        |------------->|                  |
      |      |        | Release     |                  |
   reserv.   |        |             |                  |
   state     |        | COPS-ODRA   |                  |
      |      |dispatch| DEC MESSAGE |                  |
      |      |message |<-------------|                  |
      +      |<-------| Install     |                  |
   <---------|  OK    |             |                  |
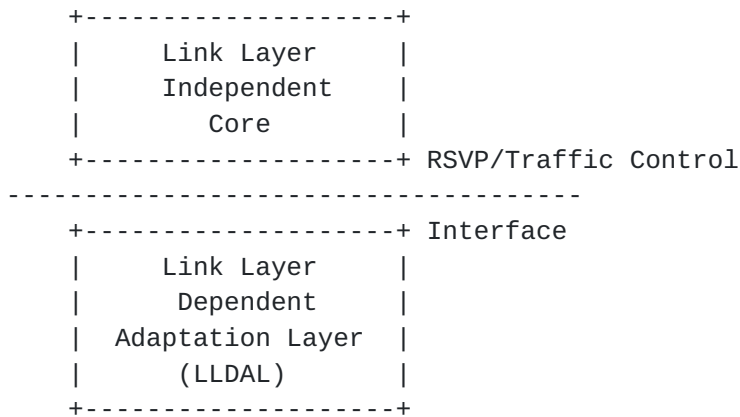     RESV TEAR|        |             |                  |
       MESSAGE...      ...          ...               ...
```

                    Figure 5: Resource release

[4](#). Traffic control in the RSVP daemon

The previous paragraph explains how RSVP and COPS can interact within
the same Edge Router. Obviously the need for interaction between them
requires some modifications in the normal operations performed by RSVP.
An example of this is represented by the introduction of pending
reservation states (see the previous paragraph). Another modification
deals with the interface between RSVP and Traffic Control, described in
[RFC2205]. As explained there, it may be desirable to organize an RSVP
implementation into two parts: a core that performs link-layer
independent processing, and a link-layer dependent adaptation layer
(LLDAL). A suitable interface between these layers is specified in
[RFC2205] (see Figure 6):

```
                    +--------------------+
                    |      Link Layer    |
                    |      Independent    |
                    |         Core       |
                    +--------------------+ RSVP/Traffic Control
             ---------------------------------------
                    +--------------------+ Interface
                    |      Link Layer     |
                    |       Dependent     |
                    |   Adaptation Layer  |
                    |        (LLDAL)      |
                    +--------------------+
```

                Figure 6: Organization of an RSVP implementation

The realization of the scenarios depicted in Figure 1 and in Figure 2
requires some minor changes to this interface. In fact, let us focus on
the TC_AddFlowspec() call:

   Rhandle = TC_AddFlowspec(Interface, TC_Flowspec, TC_Tspec,
                             TC_Adspec, Police_Flags)

In a pure RSVP/IntServ scenario, whenever the upper layer executes a
TC_AddFlowspec() call, the LLDAL performs Admission Control and, in case
of success, it sets up the reservation (i.e. it adds the proper traffic
control objects, such as schedulers and classifiers, on the interface
involved in the reservation). This makes sense, since in this case
Admission Control is based on the local availability of resources, whose
usage is likely to be known at the Link Layer dependent level.

The situation is slightly different when dealing with the interoperation
between IntServ and DiffServ. In fact, in this case Admission Control
requires the knowledge of the resource usage of the whole DiffServ
domain, and such an information is obviously outside the local scope of
the LLDAL. The entity that is aware of the whole domain is the Bandwidth
Broker, and this implies that it must be involved in the decision, i.e.
it must be queried by the RSVP. There are obviously two possible choices
to perform this operation. The first one is to keep the Admission
Control functionality at the LLDAL level, making it responsible of
interfacing with the COPS-ODRA PEP; the second one is to move this
functionality at the upper layer, thus realizing RSVP/COPS
intercommunication in the Link Layer Independent Core. The first choice
is preferable, since it leaves unchanged the semantic of the original
TC_AddFlowspec() call and complies with the philosophy behind the
scenario of Figure 2. In fact, from the RSVP viewpoint the DiffServ core
network can be seen as a single virtual trunk, i.e. a sort of virtual
link layer.

Note however that a new piece of information is required for Admission
Control in the situation depicted above. In fact, since the availability
of resources concerns a network, and not a single link, it should also
take into account some topological information, i.e. typically the IP

addresses of the ingress and egress Edge Routers. The former is known,
because it coincides with the IP address of the ingress router involved
in the operation. The latter, instead, should be retrieved from the RESV
message and passed through the interface of Figure 5 for proper
Admission Control. This would imply a change of the TC_AddFlowspec()
call, that would become:

    Rhandle = TC_AddFlowspec(Interface, TC_Flowspec, TC_Tspec,
                              TC_Adspec, Police_Flags, Egress_Point)

where Egress_Point contains the IP address of the egress Edge Router.

Figure 7 summarizes the considerations explained in this paragraph. It
depicts the situation in the Edge Router ER, where the RSVP daemon and
the COPS-ODRA Client run as independent and concurrent processes. As
mentioned before the communication between them is realized by an API,
described in a companion document [CCAPI], that is used to support
communication at the LLDAL level.

```
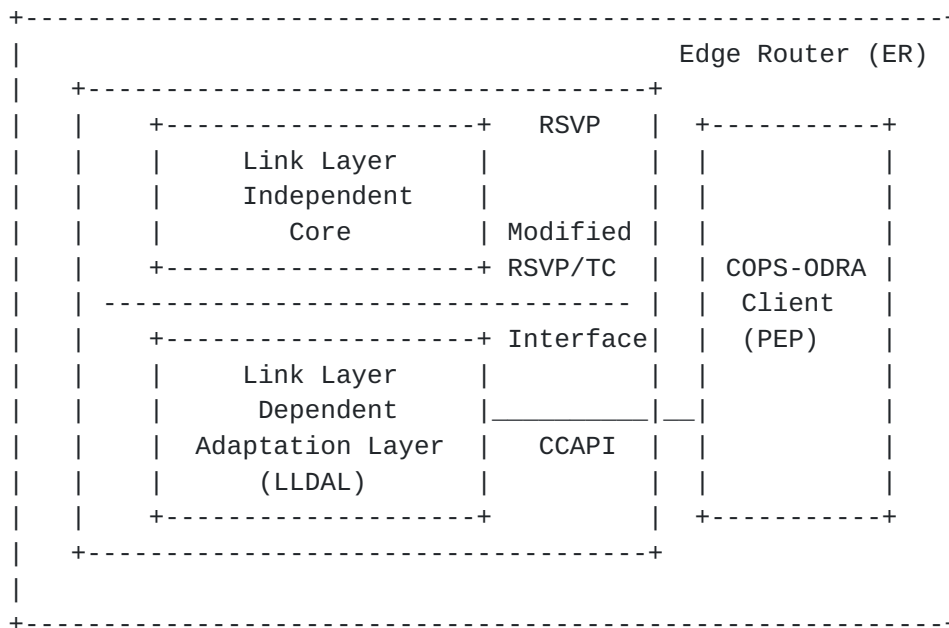    +------------------------------------------------------------+
    |                                            Edge Router (ER) |
    |   +-----------------------------------+                     |
    |   |      +--------------------+  RSVP  |  +-----------+      |
    |   |      |    Link Layer      |        |  |           |      |
    |   |      |    Independent     |        |  |           |      |
    |   |      |       Core         | Modified |  |           |      |
    |   |      +--------------------+ RSVP/TC |  | COPS-ODRA |      |
    |   | --------------------------------- |  |  Client   |      |
    |   |      +--------------------+ Interface|  |  (PEP)    |      |
    |   |      |    Link Layer      |        |  |           |      |
    |   |      |    Dependent       |_____|__|           |      |
    |   |      | Adaptation Layer   |  CCAPI  |  |           |      |
    |   |      |      (LLDAL)       |        |  |           |      |
    |   |      +--------------------+        |  +-----------+      |
    |   +-----------------------------------+                     |
    |                                                            |
    +------------------------------------------------------------+
```

            Figure 7: RSVP/COPS interaction in the Edge Router

5. References

[INTSERV]   R. Braden, D. Clark, S. Shenker, "Integrated Services in
              the Internet Architecture: an Overview", IETF RFC 1633, June
              1994
[RFC2210]   J. Wroclawski, "The Use of RSVP with Integrated
              Services", IETF RFC 2210, September 1997

[RFC2205]   R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin,
              "Resource ReSerVation Protocol (RSVP) - Version 1 Functional
              Specification ", IETF RFC 2205, September 1997
[2BIT] K. Nichols, V. Jacobson, L. Zhang "A Two-bit Differentiated
              Services Architecture for the Internet", IETF RFC 2638, July
              1999
[DSARCH]S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss,
              "An Architecture for Differentiated Services", IETF RFC 2475,
              December 1998
[INTDIF]Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer,
              R. Braden, J. Wrocklaski, E. Felstaine, "A Framework for
              Integrated Services Operation Over DiffServ Networks", IETF
              <draft-ietf-issll-diffserv-rsvp-03.txt>, September 1999, Work
              in Progress
[IWQOS99]   O. Schelen, A. Nilsson, J. Norrgard, S. Pink,
              "Performance of QoS Agents for Provisioning Network Resources",
              Proceedings of IFIP Seventh Internation Workshop on QoS
              (IWQoS'99), London, UK, June 1999
[COPS] D. Durham, Ed., J. Boyle, R. Cohen, S. Herzog, R. Rajan, A.
              Sastry "The COPS (Common Open Policy Service) Protocol", IETF
              RFC 2748, January 2000
[COPS-ODRA]   S. Salsano, "COPS Usage for Outsourcing Diffserv
              Resource Allocation", <draft-salsano-issll-cops-odra-00.txt>,
              February 2000, Work in Progress
[CCAPI] R. Mameli, "The CCAPI (COPS Client Application Programming
              Interface", <draft-mameli-issll-cops-api-00.txt>, February
              2000, Work in Progress

6. Author Information and acknoledgements

Stefano Salsano
CoRiTeL consortium
Via di Tor Vergata 135
00133 _ Roma (Italy)

Phone: +39 06 20410029
EMail: salsano@coritel.it

Roberto Mameli
CoRiTeL consortium
Via di Tor Vergata 135
00133 _ Roma (Italy)

Phone: +39 06 20410038
EMail: mameli@coritel.it