QoS Control by Means of COPS to Support SIP-Based Applications

Stefano Salsano, DIE, Università di Roma "Tor Vergata"
Luca Veltri, CoRiTeL, Research Consortium on Telecommunications, Italy

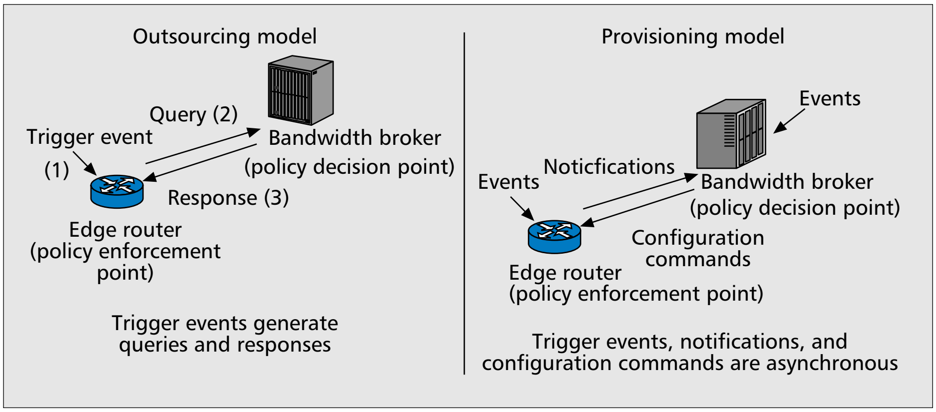
Abstract

The COPS protocol has been designed to enable communication on the interface between the policy decision administrator and the policy enforcement devices in a policy-based networking environment. It can be recognized that on the same interface there is the need to transfer information related to the request of resource by QoS clients and for the allocation of resources by resource allocation servers (e.g., bandwidth broker) in a DiffServ network. Hence, it is sensible to add this resource allocation functionality in the COPS framework. In particular, there are at least two cases where it is sensible to use COPS. The first case is on the interface between an edge node and a resource control node for handling resource allocation in a network provider domain. The second case is on the interface between a customer (client of a QoS enabled network) and the network provider: here COPS can be used as a protocol to signal dynamic admission control requests. In this article we present the definition of a new COPS client type to support the above-mentioned functionality, then describe an application scenario where SIP-based IP telephony applications can use Diffserv-based QoS networks. Simple backward-compatible enhancements to SIP are needed to interact with COPS/Diffserv QoS. A testbed implementation of the proposed solutions is finally described.

The Common Open Policy Service (COPS) protocol has been defined in the context of the IETF RAP working group as mean to support policy control in an IP quality of service (QoS) environment. The underlying architectural model foresees that policy servers administrate the network communicating decision to policy clients (e.g., network elements) where the policy decisions are enforced. Basically the decisions concern who is authorized to access what resource in the network. In particular, if IP QoS is deployed, the users can access different transport services, and this access must be administratively regulated. Two architectural models for IP QoS have been proposed in the Internet Engineering Task Force (IETF): the integrated services (IntServ) and differentiated service (DiffServ) architectures. A very good introduction to the topic of IP QoS can be found in [1], where the two approaches are described and compared. Due to scalability issues of the IntServ model, the DiffServ model seems preferable for the development of IP QoS in a real-life network. A fundamental difference between the two models is that the IntServ model includes the definition of a signaling mechanism and an admission control framework. The QoS clients use this signaling mechanism (Resource Reservation Protocol, RSVP) to express their QoS requests to the network. The network is able to properly fulfill the QoS requests using a clearly defined admission control framework. The DiffServ model still lacks a standard definition of signaling mechanisms and an admission control framework. Therefore, the currently available model for resource handling is based on a semi-static assignment of resources to a QoS client according to long-term agreements (called service level agreements) between customer and network provider. There is no dynamic signaling of requests from users to network, and it is possible to change the assignment of resources only on a relatively long timescale (e.g., days). The process of properly configuring the elements in the provider network is called DiffServ resource *provisioning*.

The concept of policy control applies to both IntServ and DiffServ networks, but the different signaling and admission control models need to be taken into account. As for the IntServ model, it is conceptually easy to add policy control on top of the signaling and admission control framework. In fact, the standardization process in the IETF followed this path: the RAP working group developed the COPS protocol with the idea to complement the *resource-related* admission control defined in the IntServ model with a *policy-related* admission control. The requirements for the initial definition of the policy-based admission control architecture and of the COPS protocol were mainly derived considering the IntServ RSVP signaling protocol. In this scenario [2] the network nodes, running RSVP, represent the policy enforcement points (PEPs), while a logically centralized element acts as a policy server and is called the policy decision point (PDP). The PEP makes *requests* to the PDP for policy-related admission control and the PDP provides the needed policy *decisions*. As for the DiffServ model, an extension to COPS to support the provisioning of resources within network elements has been defined, called COPS-PR [3]. Basically, it supports the static provisioning model discussed above. A kind of logically centralized management center acts as the PDP and "installs" the proper con-

This work has been partly supported by the European Commission under project AQUILA (IST-1999–10077.



■ Figure 1. *Outsourcing and provisioning models in COPS.*

figuration (*decisions*) in the DiffServ network elements (routers) that represent the PEPs.

The static provisioning model for a DiffServ network, although interesting in an early deployment phase for its simplicity, has some annoying limitations. For example, the preconfiguration of network elements may lead to underutilization of resources; it is difficult for the provider to adapt to changes in traffic demand; the service offering of the provider is basically limited to the transfer of large and stable traffic aggregates. The evolution of the DiffServ model envisages the capacity to dynamically handle resource requests. A higher utilization of network resources and the possibility of offering more advanced services are some of the benefits of the dynamic resource allocation.

In this article we describe a dynamic DiffServ resource allocation model that relies on COPS as a signaling mechanism (the detailed protocol specification is given in [4]). The COPS protocol provides the opportunity to combine policy control, QoS signaling and resource control in a unified framework. The model is applied to a realistic SIP-based IP telephony scenario. The SIP protocol is the IETF standard for IP telephony; it seems to be the more promising candidate for call setup signaling for the future IP-based telephony services, and it has been chosen by Third Generation Partnership Project (3GPP) as the protocol for multimedia application in 3G mobile networks. In this context, we describe a very simple solution that binds the SIP signaling to the proposed COPS-based QoS model. The SIP protocol is enhanced to convey QoS related information, preserving backward compatibility with current SIP applications and decoupling as much as possible SIP signaling from QoS handling [5]. The proposed solution fulfills the requirement of QoS support in SIP-based IP telephony.

We explain the role of COPS for resource allocation in a dynamic DiffServ context, while we discuss the definition of the COPS client type (COPS-DRA) suitable for this scenario. The QoS model is applied to the SIP-based IP telephony scenario. An implementation of the overall architecture is described, and then we draw conclusions.

The COPS Role for Dynamic DiffServ Resource Allocation

The COPS protocol is a simple query and response protocol that allows policy servers (PDPs) to communicate policy decisions to network devices (PEPs). In order to be flexible, the COPS protocol has been designed to support multiple types of policy clients. Each client type is described in a different usage document. The protocol uses TCP to provide reliable exchange of messages. COPS provides the means to establish and maintain a dialogue between the client and the server and to identify the requests. Two main models are supported by the COPS protocol: outsourcing and provisioning (Fig. 1). Under the outsourcing model, external events in the PEP (e.g., an admission control request) must be handled with a policy decision. The PEP delegates this decision to the PDP with an explicit *Request* message. The PDP makes the policy decision and answers with a *Decision* message. Under the provisioning (also known as configuration) model, the PDP proactively sends *Decision* messages to configure the resource handling mechanisms in the PEP. In other words, the network elements are preconfigured, based on policy, prior to processing events.

Let us now consider the dynamic scenario for DiffServ QoS. Two components must be defined: a signaling mechanism and an admission control framework. As for the latter, a straightforward solution is to use a server to control the admission of traffic within a DiffServ domain. This approach has been considered since the very beginning of the discussion about the DiffServ architecture [6]. The admission control server in the DiffServ terminology is typically referred to as the bandwidth broker (BB). In the dynamic scenario there is a need to exchange resource allocation requests from the edge routers to the logically centralized BB, and this can be somehow mapped into a PEP-to-PDP relationship. The commonality between the BB and the PDP has been described in [7] by the BB group in the Internet2 Qbone project. In [7] the use of COPS for the communication between the edge cevice and the BB was listed as a possible candidate for the intradomain scenario.[1] The use of COPS for dynamic resource admission control in a DiffServ network is also assumed in some studies and prototypes [8]. Anyway, no formal description of the COPS extensions for the specific scenarios has been given. We have defined the COPS extensions for DiffServ resource allocation under a strict Outsourcing model in [9]. An application scenario was described in [10]. The work is extended here with a new COPS client type that combines the Outsourcing and Provisioning model for DiffServ resource allocation.

The other component of a dynamic model for DiffServ QoS is the signaling mechanism that allows the QoS clients to make resource reservation requests to the network. A proposed solution is the use of the IntServ RSVP as the access signaling protocol [11]. We have also deeply examined this solution in [10, 12]. RSVP was designed as an end-to-end protocol to support multicast sessions spanning the whole Internet with receiver-oriented reservations. RSVP provides not only the access signaling, but all the mechanisms to enable resource reservation in the router, and it includes a lot of features like protection from routing changes, receiver diversity, and so on. We believe that using RSVP only as the access signaling protocol introduces unneeded complexity, so this solution is useful only in a potential interworking scenario where a large number of hosts (and applications) natively use RSVP.

[1] *In this article we focus on the role of the BB in controlling resources for a network in a single administrative domain (intradomain scenario). Bandwidth brokers may play also an important role in interdomain resource negotiation. The work of the Internet2 Qbone has actually focused on the interdomain aspects.*

This does not seem to be the case right now. Other solutions have been proposed based on proprietary signaling mechanisms. For example, the European IST project AQUILA [13] developed a mechanism based on a distributed object computing platform (CORBA). There is actually a need for a more systematic approach to address this problem. A commonly agreed definition of the logical content of a dynamic resource reservation in a DiffServ network is still lacking. Some efforts are underway to achieve a formal definition of the so-called service level specification (SLS), which should represent the answer to this need [14]. COPS is listed among the candidate protocols to transport SLS. In this work we make a detailed proposal on how to use COPS to signal dynamic admission control requests between the QoS clients and the provider of a QoS-enabled network.

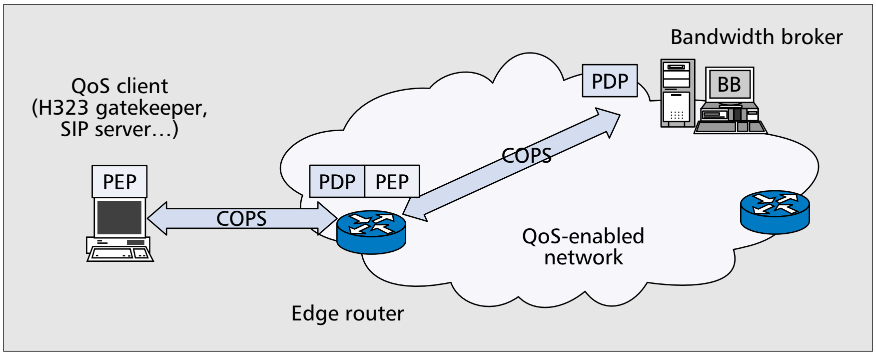


Figure 2. *COPS support to dynamic DiffServ-based IP QoS.*

Figure 2 provides a representation of the proposed architecture for dynamic DiffServ QoS. A proper extension of the COPS protocol is used on both the interface between the edge router and the logically centralized admission/policy control server, and the interface between the QoS client and the network. In the figure the QoS client is represented by a server for IP telephony, since one of the main applications of IP QoS can be the convergence of voice and data on an IP transport network. In the picture the leftmost interface is a user–network interface. The architecture can easily support other scenarios where the QoS client belongs to the provider network (e.g., a SIP server in a 3G mobile network).

Definition of the COPS Interfaces

This section describes the proposed extensions to COPS for the support of a dynamic DiffServ QoS scenario. A new client type is defined, COPS DiffServ Resource Allocation (COPS-DRA). The COPS client type defined in [9], Outsourcing DiffServ Resource Allocation (COPS-ODRA), was based only on the outsourcing model. In order to achieve a flexible and efficient network, the combination of the outsourcing model with the provisioning model should be exploited. Hereafter, the merits of the combination of outsourcing and provisioning models are discussed, comparing COPS-DRA with COPS-ODRA, and with the IETF proposed COPS-PR.

In the COPS-ODRA outsourcing model the PEP always explicitly asks the PDP/BB[2] for a given amount of resources, from an ingress point to an egress point. For scalability, per-flow state is not stored in the PDP/BB: the PDP/BB does not record each single request. Instead, resource allocation requests are properly aggregated, and only aggregate state information is kept in the PDP/BB. An example application scenario for the COPS-ODRA is IntServ operation over DiffServ networks, as described in [10]. In this scenario, the edge router includes the PEP and interacts with the PDB/BB using COPS-ODRA according to the reservation requests coming from RSVP messages. An architectural definition and scalability analysis of this scenario can be found in [15]. Due to the aggregation of state information, the COPS-ODRA model achieves good scalability in state storage, but the number of signaling messages does not scale well, for example, if a large number of relatively small requests should be supported.

A model based on provisioning is much more scalable with respect to signaling: there is no exchange of signaling messages related to single requests. The COPS-PR client type realizes this model: the PDP installs configuration decisions so that the client is able to handle events locally. The drawbacks of this model lay in inflexibility: it is difficult to handle modification of configured parameters in response to events like resource requests (each modification is handled as a request for a new configuration). It is even more difficult to handle a single "special" incoming request with the help of the PDP/BB. If fact, COPS-PR is a very general and powerful mechanism based on the use of policy information base (PIB) information. Therefore, it is able to *provision* any kind of policy in the PEP, but is not explicitly customized to handle *dynamic* DiffServ QoS.

As for efficiency, in general the outsourcing approach is more efficient in resource usage, because the resources can be allocated dynamically. The preallocation mechanisms in the provisioning approach can lead to underutilization of resources.

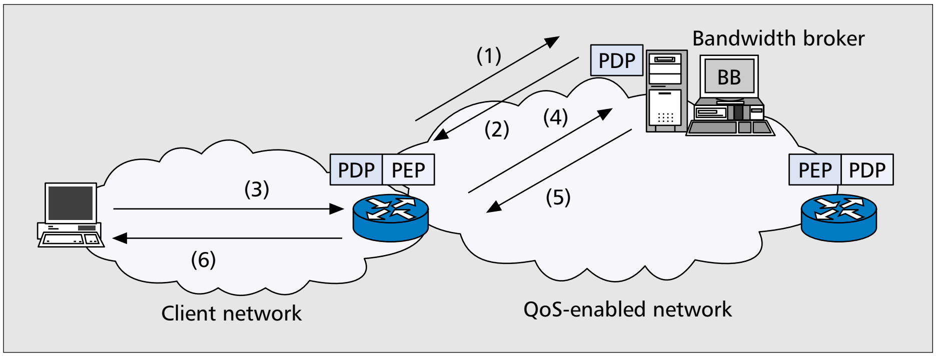
From this analysis we derive the following three requirements for a combined model:
- It should offer the capability of provisioning resources to local nodes, in order to avoid high signaling burden.
- It should be easy for the local node to request the modification (increase, decrease) of the provisioned resource.
- It should be possible to handle specific requests under the outsourcing model.

As an example, an access node to a DiffServ-enabled network with dynamic QoS will handle requests for low-bandwidth telephony calls in an "aggregated" way, based on resources provisioned by a central network management system. The access node may react to changes in the offered traffic by requesting more (or less) resources to the management system. When a request for a high quality audio and videoconference is received, the access node requests explicit admission control to the management system. A pre-allocation of resources for this kind of requests would have resulted in under-utilization of network. We have mapped these requirements into the COPS-DRA protocol, where PEP represents the access node and the PDP/BB represents the management system. COPS-DRA is explicitly targeted to DiffServ resource allocation and admission control.

The above discussion is mainly related to the communication between the access node and the logically centralized server in a dynamic DiffServ QoS scenario. Let us consider the interface between a QoS client and the QoS provider in a QoS-enabled network. On this interface the QoS client sends QoS reservation requests to the provider who is in charge of accepting or rejecting the request. In a typical scenario, considering that the provider will not distribute resources in advance to its clients, only the outsourcing model is relevant.

Once the requirements about the general model have been

[2] *The BB that combines the policy control and admission control is denoted PDP/BB.*



■ Figure 3. *An example information exchange using COPS-DRA.*

	Resource allocation model	Component of request messages
Edge router to PDB/BB interface	Outsourcing and Provisioning	Scope and amount of reservation Type of service
QoS client to QoS provider interface	Outsourcing only	Flow identification Scope and amount of reservation Type of service

■ Table 1. *COPS-DRA usage on the two different interfaces.*

considered, the logical content of the request messages sent by the PEP to the PDP should be discussed. The three basic components of the reservation requests are:
- The *scope* and *amount* of reservation (where the reservation applies and how much bandwidth)
- The *type of* requested *service* (possibly including a set of QoS parameters)
- The *flow identification* (i.e., to which IP flow or aggregate of flows the reservation applies)

The first two components are needed on both the edge router to PDP/BB interface and the QoS client to QoS provider interface, while the third component is only needed in the latter interface. As already mentioned, work is ongoing to propose a commonly agreed on definition of the semantic content of reservation requests, but this work is in a very early stage [14]. For COPS-DRA a simple scenario has been considered in order to derive the requirements:
- The scope of the reservation is identified by an ingress and an egress point in the QoS-enabled network, and the amount of needed resource is identified by a bit rate (bits per second).
- The type of requested service is simply an index to a previously agreed on list of services.
- For the flow identification the source and destination IP addresses and TCP/UDP ports are used.

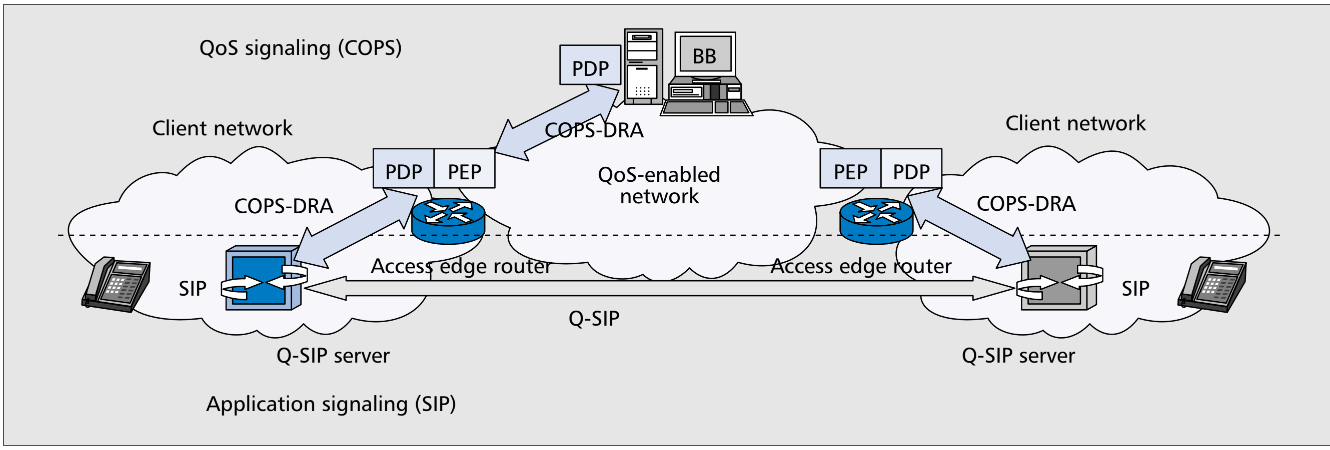
More complex scenarios may require the addition of specific parameters to these components or the definition of further components in the reservation request. For example, the “timing” of a reservation (immediate reservation, advance reservations and so on) has not been considered in the three mentioned components. Note that thanks to its extensibility, further functionality may be added to COPS-DRA in a later stage. Table 1 lists the different features of COPS-DRA used on the two different interfaces.

Hereafter, the proposed model is explained with the help of the example information exchange depicted in Fig. 3. The QoS user (e.g., a SIP proxy server as described in the next section) implements a COPS-DRA client, while the edge router plays the role of QoS provider and implements a COPS-DRA server. The edge router receives the QoS request (3) from the QoS user; then it has to reply (6). If the answer is positive it has to set up the needed elements in the forwarding path (classifier, policer, shaper). For scalability reasons the edge router is able to answer most of the resource request without contacting the PDP/BB. In a preliminary phase, always using the COPS-DRA, the edge router requires an initial configuration (1) to the PDP/BB that allocates a set of resources to the edge router (2). A specific request may require an external decision (e.g., when the bandwidth exceeds a prefixed limit). In this case the decision is outsourced to the PDP/BB, and the message (4) is the logical copy of message (3). Therefore, the answer (6) to the QoS client will be the logical copy of the PDP/BB answer (5). A different situation is when the request coming from the user can be aggregated with previous requests, but there are no more available allocated resources. In this case the edge router will send a request (4) to increase the resources allocated to the PDP/BB, which will answer with decision (5). Typically the (4) and (5) messages are sent asynchronously with respect to (3) and (6), when the edge router detects that the available allocated resources are below some thresholds. The set of edge routers and the BB realize a sort of distributed BB in a DiffServ network. The PDP/BB could base its operation on static information or interact with a network device to acquire network topology information and even to configure nodes. A hierarchy of PDP/BB and redundant elements can be used. The definition of these mechanisms and of the algorithms used by the PDP/BB to make admission control decisions are outside the scope of this article. Examples of actual algorithms for (re) distribution of resources between PDP/BB and edge routers can be found in [13, 16].

The signaling details of the COPS-DRA are fully specified in [4], only some comments to Fig. 3 are given hereafter. Messages (1), (3), (4) are COPS *Request* messages, while (2), (5), (6) are COPS *Decision* messages. Different types of *Request* messages are discriminated by a field in the so-called *Context* object. In particular, message (1) is the initial configuration request sent by the PEP to the PDP. Messages (4) can be outsourcing requests or requests related to aggregated resources: the above mentioned field discriminates the two cases. In the typical scenario, messages (3) are always outsourcing requests. The *Client Specific Info* object in the *Request* messages has been defined to carry the needed information. For example, in messages (4) the *Client Specific Info* will carry the ingress and egress point of the reservation, requested bandwidth, and type of service. In messages (3) the *Client Specific Info* will also contain the flow identification details.

IP Telephony: A COPS Based QoS Model

The SIP protocol has been defined within the IETF as a signaling protocol to initiate voice, video, and multimedia sessions, and it is a candidate for call setup signaling in IP telephony. Obviously, for the realization of such service it is very important to bind the call setup procedures with QoS reservation and/or admission control mechanisms. In the recent past, different scenarios have been proposed in order



■ Figure 4. *QoS SIP architecture.*

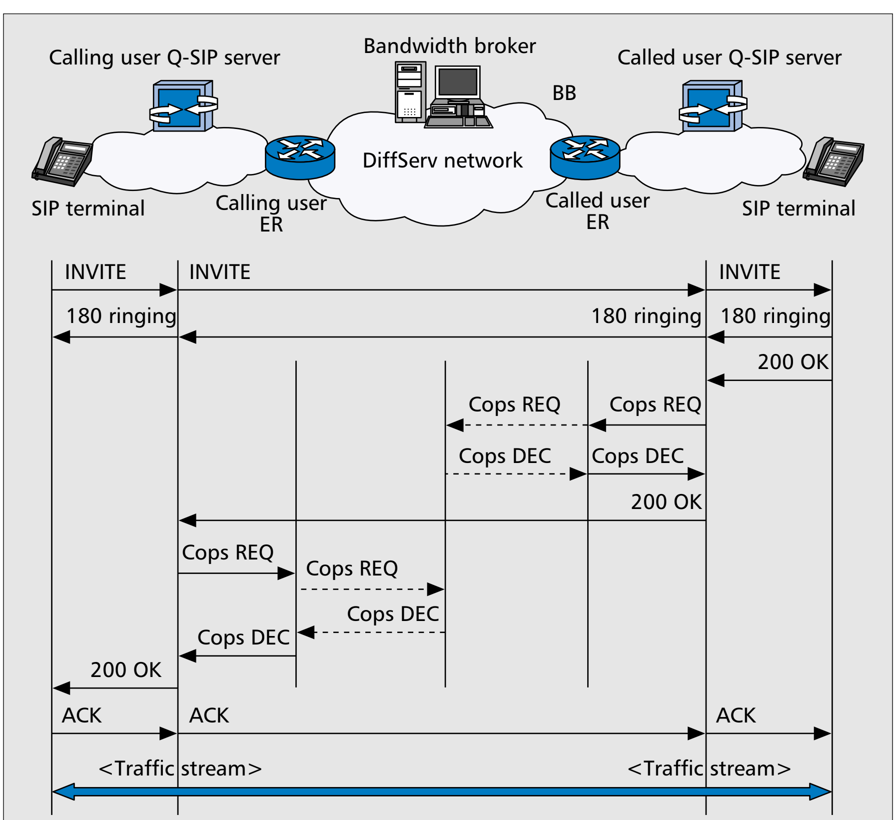
to bind the SIP signaling to the IP QoS mechanisms, but unfortunately they only refer to IntServ-based approaches [17]. These proposals consider the terminals aware of the implemented QoS model and let the terminal in the caller's network request QoS of the network. This approach has various drawbacks. First of all, the clients should be customized with the QoS mechanism used in the network, making existing legacy applications unusable. Another problem is that the terminal should implement the complete stacks for both call setup protocol (SIP) and QoS reservation. This may be a burden for "light" terminals with limited memory and processor capacities such as mobile phones or other handheld IP-based terminals.

The above considerations are the basis of the QoS architecture proposed here. The main idea is to eliminate the need for a specific QoS protocol in the terminals, and to use SIP as the sole call setup protocol for both QoS and non-QoS calls. An additional advantage is that all the QoS-related functions can be moved from the terminal to local SIP proxy servers that will control both call setup and resource reservation, thus relieving the terminals of unneeded complexity and preserving backward compatibility with standard SIP clients.

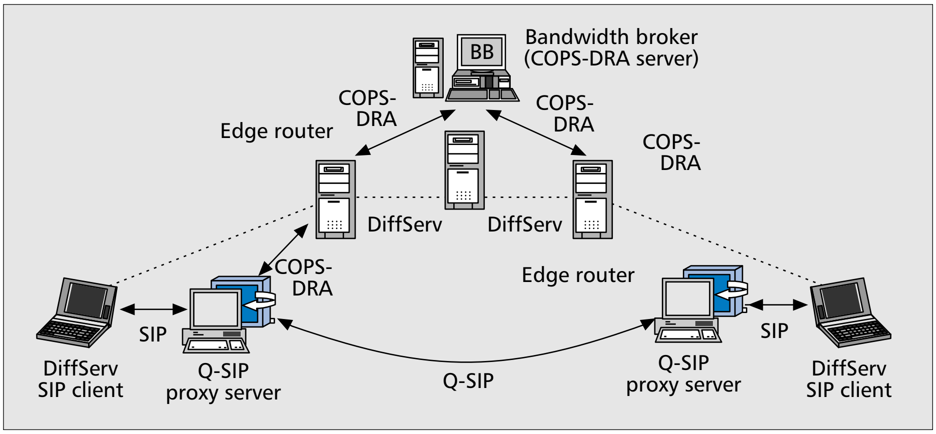
The QoS requests are handled at the border of the core network by the edge routers (ERs) that implement all mechanisms needed to perform admission control decisions (possibly with the aid of a BB) and policing function, as described in the previous sections. The COPS protocol is used to make QoS reservation requests to the QoS access points (i.e., to the network ERs). In this scenario the SIP clients are assumed to use a default SIP proxy server in their domain for both outgoing and incoming calls. The SIP servers are therefore involved in the message exchange between the clients and can add (and read) QoS related information in the SIP messages. This QoS information exchange can be made transparent to the clients. The SIP servers will negotiate QoS parameters among them and interact with the network QoS mechanisms. For the setup of a bidirectional QoS communication, two different reservations have to be requested of the QoS network. The enhanced SIP server is called a Q-SIP server (QoS-enabled SIP server). A detailed description of the Q-SIP protocol can be found in [5].

The reference scenario is depicted in Fig. 4. The involved actors are the two SIP clients, the two SIP servers and a QoS-enabled network. In the end-to-end signaling route, one or more non-QoS-aware SIP servers can be encountered, without changing the reference model.

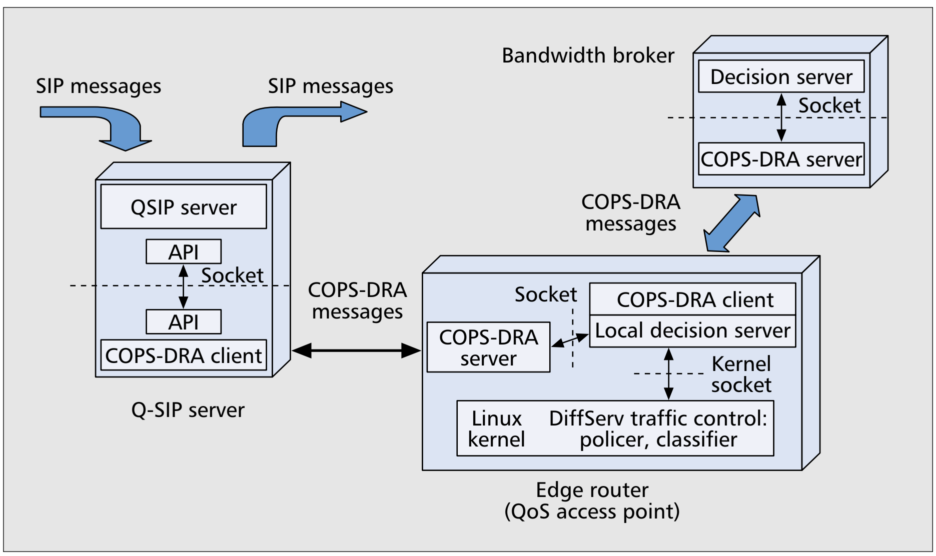
A high-level description of the architecture and of the signaling flow is given hereafter; a detailed definition is given in [5]. When a call setup is initiated, the calling user SIP client starts the SIP call setup procedure through the SIP proxy server. If a Q-SIP server is encountered, it will start a QoS session interacting with a remote Q-SIP server and with the QoS provider for the backbone network (i.e., the access ER). QoS-related information is added to SIP messages using the *VIA* fields in a backward-compatible



■ Figure 5. *Q-SIP call signaling flow.*



■ Figure 6. *The overall testbed scenario.*



■ Figure 7. *Q-SIP server, ER, and BB internal architectures.*

way. This information is transported transparently by legacy QoS-unaware SIP servers and clients. Figure 4 shows the architecture, while Fig. 5 describes the call setup message flow.

With reference to Figs. 4 and 5, the calling user sends a standard SIP *INVITE* message to the local Q-SIP server. The calling user sees the Q-SIP server as a standard SIP proxy server. The Q-SIP server, based on the calling user identity and session information, decides whether a QoS session has to be started or not. If a QoS session has to be set up, it inserts the required QoS session descriptors within the *INVITE* message and forwards it toward the invited called user; the *INVITE* messages can be relayed by both standard SIP proxy servers and Q-SIP servers. When the *200 OK* response reaches the Q-SIP server that controls the called user client network, the Q-SIP server starts a QoS reservation (COPS REQ message) to the called user access ER for the called-user-to-calling-user traffic flow. When the called user Q-SIP server receives the COPS DEC message, it sends the *200 OK* message with the complete QoS session descriptors to the calling user. When the calling user Q-SIP server receives the *200 OK* message, it completes the QoS session setup sending a QoS request to the calling user ER for the caller-to-callee traffic flow. Once the caller Q-SIP server receives the COPS DEC message in response, it forwards the *200 OK* message to the calling user. Finally, the SIP acknowledgment (ACK) traverses the network from the calling to the called user, and data transfer begins.

Implementation Testbed

The proposed architecture has been implemented in a testbed showing both QoS and call setup aspects. The testbed is composed of two Ethernet based client networks and a DiffServ core network composed of two ERs and one core router. In each access network there are one SIP terminal and one Q-SIP server. A PDP/BB node is located in the DiffServ network. The overall picture of the testbed is shown in Fig. 6.

All nodes (client, servers, and routers) are based on Linux OS. The terminals implement standard SIP clients, while Q-SIP, BB, and ERs implement ad hoc COPS clients/servers. The source code is available under the GPL license in [4]. Further details on the DiffServ implementation can also be found in [12], while [10] describes the testbed of the previous COPS-ODRA client. The Q-SIP server includes a SIP server developed in Java and a COPS-DRA client developed in C. The SIP server and COPS-DRA client are two different UNIX processes communicating through a socket interface. The ERs act as QoS access points and include a COPS-DRA server that communicates through a socket interface with a process implementing the local decision server and the COPS DRA client. This process communicates with the DiffServ traffic control mechanism provided by the Linux kernel. The PDP/BB is composed of a COPS-DRA server and a decision server that interact through a socket-based interface. Figure 7 shows the internal architectures of the Q-SIP server, ER, and BB.

We have tested in our testbed some publicly available SIP clients, and they were able to correctly dialog with our Q-SIP servers, which were in turn able to successfully set up and release QoS reservations using COPS-DRA. Future work in the testbed will address the evaluation of "post-dialing" setup delays as a function of signaling load. In the current unloaded experiments additional post-dialing setup delay due to COPS-DRA resource reservation was not perceptible by the user, but of course this result is of limited significance.

Conclusions

In this article the support of dynamic resource allocation and policy control in DiffServ-based networks has been considered. We focus on the signaling mechanism needed to handle:

- Resource admission control within a DiffServ domain
- Resource requests to a QoS provider
- QoS-aware call setups for SIP-based applications

For the first two cases, a homogenous scenario has been individuated, in which the COPS protocol is chosen as common signaling mechanism, combining QoS signaling and resource admission control. These aspects complement the defined use of COPS as policy control mechanisms.

As for the resource admission control within a DiffServ domain, the PEP is the node that handles resource and policy enforcement (typically the edge router), while the PDP is the server that handles resource allocation and policy decisions (a bandwidth broker in DiffServ terminology).

As for the resource requests to a QoS provider, the PEP is a QoS client or call control server (e.g., an H.323 gatekeeper or SIP server) that asks for QoS reservations to a QoS access point (e.g., edge routers of the DiffServ network) acting as the PDP.

A COPS client type (COPS-DRA) has been defined in order to support the combination of the two different COPS resource allocation models: outsourcing and provisioning. So far there has been no proposal of such combined mechanisms. The combination of the two models results in a flexible and efficient solution. Thanks to the provisioning component, it is possible to control the number of exchanged messages obtaining good scalability performance. Explicit requests can still be handled, applying the outsourcing model, achieving an efficient usage of resources. The typical strategy is that small requests are handled in an aggregated way (provisioning), large requests on a single flow basis (outsourcing).

As for the QoS-aware call setups for SIP-based applications, a new scenario integrating the SIP signaling with the DiffServ QoS mechanisms has been considered. We have proposed a very simple architecture in which the SIP signaling is bound to the proposed COPS-based QoS model leading to a very light and powerful model. The proposed solution enhances the SIP protocol to convey QoS-related information, but preserving backward compatibility with current SIP-based equipment that does not support QoS. This allows a smooth migration and is very interesting, for example, in the context of a third-generation mobile network, where SIP has been chosen as the protocol for the multimedia domain. A possible deployment scenario based on QoS-aware SIP proxy servers is proposed, having the advantage that legacy SIP user applications can be fully reused. The testbed implementation of the proposed solution, including the internal architecture of the Q-SIP proxy server has been described.

In conclusion, the described QoS scenario represents a promising solution for the evolution of DiffServ QoS from a static approach to a dynamic one. The dynamic approach will answer the need of present and future QoS-sensitive services in a much more efficient and flexible way.

References

[1] X. Xiao and L.M. Ni "Internet QoS: A Big Picture," *IEEE Net.*, Mar. 1999.

[2] J. Boyle *et al.*, "COPS Usage for RSVP," IETF RFC 2749, Jan. 2000.

[3] K. Chan *et al.*, "COPS Usage for Policy Provisioning," IETF RFC 3084, Mar. 2001.

[4] S. Salsano "COPS Usage for DiffServ Resource Allocation (COPS-DRA)," <draft-salsano-cops-dra-00.txt>, Oct. 2001, work in progress, http://www.coritel.it/projects/cops-bb

[5] L. Veltri and S. Salsano, "QoS Support for SIP Based Applications in DiffServ Networks," <draft-veltri-sip-qsip-00.txt>, Oct. 2001, work in progress, http://www.coritel.it/projects/qsip

[6] K. Nichols, V. Jacobson, and L. Zhang "A Two-bit Differentiated Services Architecture for the Internet" RFC 2638, July 1999.

[7] R. Neilson *et al.*, eds., "A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment" Version 0.7.

[8] First Internet2 bandwidth broker operability event http://www.merit.edu/internet/working.groups/i2-qbone-bb/inter-op/index.htm

[9] S. Salsano, "COPS Usage for Outsourcing DiffServ Resource Allocation," <draft-salsano-cops-odra-01.txt>, Sept. 2001, work in progress, http://www.coritel.it/project/cops-bb

[10] R. Mameli and S. Salsano "Use of COPS for IntServ Operations over DiffServ: Architectural Issues, Protocol Design and Testbed Implementation," *ICC 2001*, Helsinki, Finland.

[11] Y. Bernet *et al.*, "A Framework for Integrated Services Operation over DiffServ Networks," IETF RFC 2998, Nov. 2000.

[12] W. Almesberger *et al.*, "A Prototype Implementation for IntServ Operation over DiffServ Networks," *IEEE GLOBECOM 2000*, S. Francisco, Dec. 2000.

[13] B. Koch and S. Salsano, "IP QoS at Work: Definition and Implementation of the AQUILA Architecture," *IWDC 2001*, Taormina, Italy Sept., 17–20, 2001, LNCS 2170, http://www.ist-aquila.org (Publications)

[14] Service Level Specification Interest Web Page, http://www.ist-tequila.org/sls.html

[15] A. Detti *et al.*, "Supporting RSVP in a Differentiated Service Domain: An Architectural Framework and a Scalability Analysis," *ICC '99*, June 1999, Vancouver, Canada, pp. 204–10.

[16] E. Nikolouzou *et al.*, "An Adaptive Algorithm for Resource Management in a Differentiated Services Network," *ICC2001*, Helsinki, Finland.

[17] W. Marshall *et al.*, "Integration of Resource Management and SIP," IETF Internet draft <draft-ietf-sip-manyfolks-resource-01.txt>, Feb. 2001, work in progress.

Biographies

STEFANO SALSANO (salsano@coritel.it) received his Laurea degree in 1994 (University of Rome "Tor Vergata") and his Ph.D. in 1998 (University of Rome "La Sapienza"). Since November 2000 he has been an assistant professor at the University of Rome "Tor Vergata." From 1997 to 2000 he was with CoRiTeL, a research institute on telecommunications, where he was coordinator of IP-related research activities. He has participated in several research projects founded by the EU (INSIGNIA, ELISA, AQUILA), the European Space Agency and the Italian Ministry of Research. His current research interests include QoS and traffic engineering in IP networks, IP telephony, MPLS, and IP over optical networks.

LUCA VELTRI (veltri@coritel.it) received his Laurea degree in telecommunication engineering from University of Rome "La Sapienza" in 1994. He received his Ph.D. in communication and computer science from the same university in 1999. Since 1999 he has been with CoRiTe, a research institute on telecommunication that joins Ericsson Lab Italy and the university. In CoRiTeL he is coordinator of research activities in IP networking. He has participated in several research projects founded by the EU, ESA, and the Italian Ministry of Research. His current research interests include QoS in IP networks, IP telephony, and IP mobility for 3G/4G networks.