A SIP-based Network QoS Provisioning Framework for Cloud-hosted DDS Applications

Akram Hakiri[1], Aniruddha Gokhale[2], Douglas C. Schmidt[2], Berthou Pascal[1], Joe Hoffert[2], and Gayraud Thierry[1]

[1] CNRS; LAAS, 7, avenue du Colonel Roche,
Universit de Toulouse; UPS, INSA, INP, ISAE; LAAS;
F-31077 Toulouse, France
[2] Institute for Software Integrated Systems, Dept of EECS
Vanderbilt University, Nashville, TN 37212, USA
Hakiri, Berthou, Gayraud@laas.fr,
a.gokhale,d.schmidt,jhoffert@vanderbilt.edu

Abstract. The growing trend towards running publish/subscribe (pub/-sub)-based distributed real-time and embedded (DRE) systems in cloud environments motivates the need to achieve end-to-end quality-of-service (QoS) over wide-area networks (WANs). The OMG Data Distribution Service (DDS) is a data-centric middleware that provides fast, scalable and predictable distribution of real-time critical data. The DDS standard, however, provides QoS control mechanisms that are confined only to the middleware residing at end-systems, which makes it hard to support DRE pub/sub systems over WANs. A promising solution to this problem is to integrate DDS with the Session Initiation Protocol (SIP), which is an IP-based signaling protocol that supports real-time applications involving voice, video, and multimedia sessions via the QoS mechanisms in IP networks.

This paper describes our approach to bridge the SIP protocol and DDS to realize DDS-based applications over QoS-enabled IP WANs by overcoming both inherent and accidental complexities in their integration. An exemplar of the proposed approach for IP DiffServ networks is described, which uses the Common Open Policy Server (COPS) protocol to assure QoS for cloud-hosted DRE pub/sub applications. To distinguish the DDS traffic from other best-effort traffic in the cloud environment, our approach uses the COPS-DRA protocol as a generic protocol for automatic service-level negotiation and the integration of this protocol in an overall QoS management architecture to manage service levels over multiple domains deploying different QoS technologies.

Keywords: Cloud, End-to-End QoS, DDS, SIP, COPS, DiffServ.

1 Introduction

Many publish/subscribe (pub/sub)-based distributed real-time and embedded (DRE) systems operate in heterogeneous environments that receive data from a large number of sensors and multimedia sources and stream it in real-time to remote entities. Examples of such DRE systems include unmanned air vehicles, video surveillance, on-demand video transmission, online stock trading, air traffic management, power grid control, shipboard computing environments, and weather monitoring. These types of DRE systems often typically optimize the performance and scalability of applications and provision/control key network resources [18].

The Object Management Group (OMG)'s *Data Distribution Service* (DDS) [14] is a data-centric pub/sub middleware that simplifies application development, deployment, and evolution for DRE systems. DDS delivers needed capabilities of DRE systems via powerful quality-of-service (QoS) policies that enable applications to control data delivery properties at each distributed node. In addition, DDS supports fast and predictable distribution of real-time critical data over heterogeneous networks.

Most applications of DDS in pub/sub DRE systems either use stable networks, such as local area networks, or networks whose scale is small and whose properties are controllable. With a growing trend [1] towards supporting pub/sub DRE systems in the cloud, however, it becomes necessary to realize the QoS mechanisms for pub/sub DRE systems in wide-area networks (WANs) that are characteristic of cloud environments. The DDS specification, however, does not currently support key QoS properties of pub/sub DRE systems over WANs because the DDS middleware resides on end-systems and thus defines only mechanisms that control end-system properties, such as OS-level parameters and tuning network parameters for the connecting link.

For example, DDS provides no mechanisms for provisioning/controlling end-to-end QoS over WANS. The lack of these mechanisms makes it hard to use DDS to assure end-to-end QoS for large-scale DRE systems running in clouds. Key challenges required to support cloud-based DDS applications include optimizing the performance and scalability of WAN deployment over fixed and wireless access technologies while also providing network-centric QoS provisioning.

A promising approach to address these challenges is to integrate DDS with the Session Initiation Protocol (SIP) [16] and Session Description Protocol (SDP) [17]. Together, SIP and SDP provide a powerful capability to convey new enhanced services. Examples of such services include disseminating information about end-systems, identifying the originator of a session, and identifying multimedia content in a session initiation request that might contain pictures, signals from sensors, or a personalized ring-tone.

Despite the strengths of SIP/SDP, they do not take into account the application needs of pub/sub DRE systems. In particular, the desired QoS policies of pub/sub DRE systems cannot be seamlessly supported today both by DDS middleware on end-systems and SIP/SDP mechanisms in WANs. To bridge the gap between DDS and SIP/SDP, therefore, this paper describes a SIP-based QoS management architecture using a proxy SIP QoS module, which implements both DDS QoS policies and standardized QoS mechanisms. Our approach defines a new SIP *Signaling Class of Service* (S-CoS) for transferring signaling messages for sessions whose characteristics are specified in extensions we defined for SDP messages. A key benefit of our approach is that it require no modifications to applications, which can continue to use standard DDS QoS policy and programming interfaces.

To demonstrate our approach, we have prototyped our QoS-enabled SIP proxy for cloud environments using DiffServ IP networks, which makes it possible for interworking DDS sessions with QoS-enabled IP networks by integrating it with DiffServ mechanisms such as the Common Open Policy Server (COPS) [10] protocol. COPS aims to exchange policy information between a policy server (Policy Decision Point or PDP) and its clients (Policy Enforcement Points or PEPs), using TCP transport protocol for reliable exchange of messages between PEPs and PDPs. In our case, therefore, COPS acts as the protocol for QoS requests and for admission control to achieve the desired QoS for DDS traffic over DiffServ IP networks.

The remainder of this paper is organized as follows: Section 2 describes how we integrated DDS with SIP/SDP to support pub/sub-based DRE systems in cloud environments; Section 3 showcases our integrated solution in a DiffServ network, focusing on the signaling procedure and QoS provisioning; Section 4 compares our research with related work; and Section 5 presents concluding remarks.

2 Supporting DDS in Cloud Environments using SIP/SDP

This section describes the SIP/SDP-based framework we developed to support DDS session in cloud environments, which comprise WANs. To better understand our solution, we first explain how applications use DDS and provide an overview of the SIP/SDP protocols. We then describe the enhancements we made for SIP/SDP media description to support DDS sessions over WANs.

2.1 Overview of Underlying Technologies

This section summarizes the DDS, SIP, and SDP technologies.

DDS and DDS sessions. DDS is a middleware standard for distributed real-time application that simplifies application development, deployment and maintenance and provides fast, predictable distribution of real-time critical data over heterogeneous networks. The DDS architecture consists of two layers. The *Data-Centric Publish Subscribe* (DCPS) layer provides efficient, scalable, predictable, and resource-aware data distribution. The *Data Local Reconstruction Layer (DLRL)* provides an object-oriented facade atop the DCPS so that applications can access object fields rather than data and defines navigable associations between objects.

The DCPS DDS entities include *topics*, which describe the type of data to read or write, *data readers*, which subscribe to the values or instances of particular topics, and *data writers*, which publish values or instances for particular topics. Properties of these entities can be configured using combinations of the DDS QoS policies.

SIP and SDP. The Session Initiation Protocol (SIP) is an IP-based application-level signaling protocol defined by the IETF in RFC3261 [16]. It plays a major role in establishing, maintaining, modifying, and tearing-down multimedia sessions between two distributed end-points. SIP focuses on IP core networks for all services: mobility, scalability, multimedia services, high bit rate, and dissemination of call/service control and user data. From the service plane (application signaling) point of view, SIP simplifies the network operation and management by automating the QoS provisioning, and providing the level of QoS by facilitating admission control with the network QoS negotiation mechanisms.

Multimedia sessions can be described using the Session Description Protocol (SDP) [12]. Information in the session defines a syntax to characterize the multimedia session, such as the types of media in the session, the available support for each of the media types, the contact information (IP address (IP/port) where packets should be sent) and bandwidth requirements for receiving the session. SDP is therefore decomposed into three main descriptors: (1) *session-level descriptions*, which describe characteristics of the whole session; (2) *time descriptions*, which indicate time-related aspects of the session, and (3) *media descriptions*, which characterize the different media present in the session.

2.2 SIP/SDP Enhancements for Cloud-based DDS

Section 1 alluded to the limitations in realizing cloud-based DDS applications. We present our SIP/SDP enhancements to support cloud-based DDS applications as shown in Figure 1. Cloud-based DDS applications are expected to be located at the access network. Specifically, SIP is used with the discovery protocol to discover end-systems while SDP is used to encapsulate DDS QoS policies.

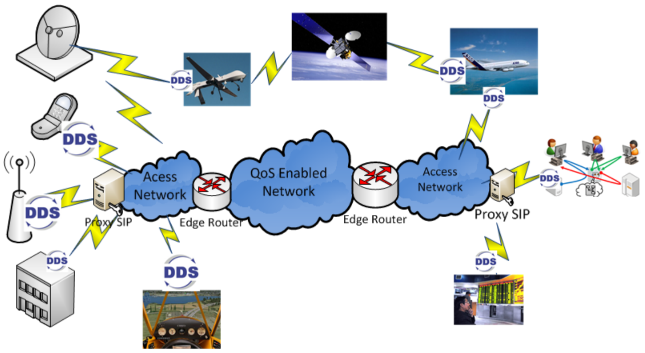

Fig. 1. A SIP-based, DDS-enabled QoS Support Architecture

We propose to let SIP messages carry DDS QoS policies, which in turn are used in reserving the required network resources. The signaling part of SIP is therefore independent of the session being established and of the mechanism used to describe it. This approach provides the way to distribute this information between potential participants in the session, *i.e.*, the QoS information exchange is made transparent to the DDS clients. The proxy SIP negotiates on behalf of the clients with the network QoS mechanisms. The DDS QoS policies are mapped into SIP/SDP messages, and carried within a new SDP media session description attribute we added to the SIP protocol stack, and used by the Proxy SIP server.

2.3 End-system Architecture

Figure 2 presents an architectural view of interfaces and functional entities that comprise our QoS support architecture, which is located both at the sender and the receiver sides.

To establish a communication between SIP applications (without DDS) and DDS applications attached to SIP entities, the following entities are required: (1) The SIP User Agent (UA) acts as an interface between the QoS-enhanced SIP Proxy (Q-SIP) and DDS application, (2) SIP Proxy which maintains media session between DDS pub/sub participants. the remainder of this section describes the specifics of the SIP Proxy and the new SDP attributes we added.

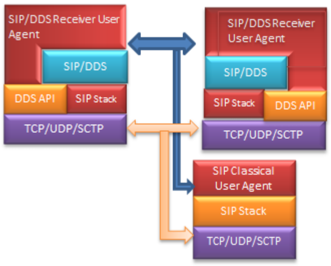

Fig. 2. DDS Mapping Interface with a SIP Stack

DDS application. DDS application are not integrated within the SIP-based environment, as shown in Figure 2. Indeed, because these application contain a rich set of domain users, it was less convenient to implement them within the SIP interface. Moreover, for extensibility reasons, to make use of general SIP applications like IMS applications, we determined the application to remain independent of the SIP/DDS implementation.

SIP user agent (UA). The UA is not integrated within the application. The UA acts as an interface between the Proxy SIP and DDS application. This approach preserves interoperability properties since any SIP-based application can communicate with a DDS-based application.

Proxy server. The goal of the proxy server architecture is to support both wired and wireless communication between DDS-based pub/sub system and other wireless access points. Moreover, DDS application can access a network via a mobile client. The Proxy SIP is considered as a network element. Hence it is included in the network vision to allow the session management.

The proxy SIP is present in all phases of the SIP-based communication including registration (to access to the registrar server matching two SIP-UA requests), establishment (forcing all SIP and SDP messages to pass across it with the header "record route"), and it is able to understand the information about the session. The memory footprint of the proxy SIP implementation is low.

2.4 Proxy SIP Signaling Mechanism

The proxy SIP receives SIP requests/responses and forwards them through the network on behalf of the user agent. Proxies are essentially routers that are also capable of generating requests and responses. Two key types of SIP proxies are supported:

- **Inbound proxies**, which handle incoming requests for an administrative domain and also routes incoming requests to the appropriate UA within the domain it is responsible for.
- **Outbound proxies**, which help the UAs to route outgoing requests. UAs are usually configured to route all their requests to an outbound proxy, which will route the requests for them. Since outbound proxies include all the capabilities of inbound proxies, in the rest of this paper we focus only on outbound proxies.

2.5 SDP Extensions

We extended SDP messages to support the media description of the DDS session. The new SDP attributes deal with the session management and are used by the SIP User Agent (UA) that we collocate with both the sender and the receiver to indicate how the QoS may be achieved. An SDP message consists of a set of lines of text of the form:

$$< attribute >=< value >$$

The offer/answer [17] model for SDP assumes that endpoints somehow establish the QoS required for the media streams they establish. For DDS session, however, more than one QoS service is available, which requires the capability to negotiate which QoS mechanism to use for a particular media stream. We therefore created new DDS QoS attributes and syntax to incorporate the signaling procedure.

Table 1 and 2 describe the enhancements to the SIP/SDP header and the new attributes we added to support the specific DDS QoS requirements to be assured by cloud environments. The caller and callee terminals exchange SIP messages that include the resource reservation demands and the current status to support the QoS demands in each direction.

Table 1. Description of the DDS QoS policy Attributes

attribute	Description
Deadline	DataReader expects a new sample updating the value of each instance at least once every deadline period. DataWriter indicates that the application commits to write new value for each instance managed by this DW at least once every deadline period. The Deadline is a duration ”0 0”.
Latency	The delay from data writing until its delivery is inserted in the receiver’s application cache and the receiving application is notified of the fact. The Latency Budget is duration ”0 0”.
Reliability	Is the reliability of the service. Could be Reliable ”R” or Best Effort ”BE”.
Priority	The Transport Priority is a hint to the infrastructure used to set the priority of the underlying transport used to send data in the DSCP field for DiffServ. This is presented as an integer value.

Table 2. New SDP Attributes for DDS-based QoS Support

attribute	value
”qos-dds”	Deadline Latency Reliability Priority

The ”qos-dds” token identifies a QoS mechanism that is supported by the entity generating the session description. A token that appears in a ”qos-dds-send” attribute identifies DDS QoS policies supported by the DataWriters to assist the resource reservation for traffic sent by the publisher generating SDP messages. A token that appears in a ”qos-dds-recv” attribute identifies the DDS QoS policies that can be supported by the DataReader to reserve the resources for traffic coming from the DDS publisher.

For example, the SDP session description presented in Table 3 offers video communication requesting a total bandwidth of 64 kilobits per second as described in line ”b” with *qos-dds-send* and *qos-dds-recv* attributes.

2.6 Semantics of SIP/SDP Extensions

Offer/answer behavior. When using *qos-dds-send* and *qos-dds-recv* attributes, an offer/answer negotiation is done between the publisher and the subscriber to allow endpoints to load a list of QoS mechanisms. The participants negotiate the direction in which the QoS mechanisms are exchanged with respect to both preconditions [2] and DDS changeable table parameters [14]. Participants may also use other QoS parameters (such as those described in [8]) to allow bandwidth,

Table 3. Example of an **m** Line with "qos-dds" Attributes

```
v = 0
o = alice 2890844526 2890842807 IN IP4 1.2.3.4
s =
c = IN IP4 1.2.3.4
t = 0 0
m = video 51372 RTP/AVP 31
b = AS:64
a=qos-dds-send: 0 0 0 0 R 12
a=qos-dds-recv: 0 50 0 5 R 12
```

jitter, delay parameters to be negotiated at the same time with per-flow resources reservation for IntServ/RSVP, and per-Class for DiffServ infrastructure.
Offer behavior. Publishers include *qos-dds-send* flow in the publication direction to inform subscribers about the QoS parameters supported by the sender. Similarly, a participant can use *qos-dds-recv* attributes to specify which kind of QoS policies can be supported at the receive direction.
Answer behavior. After receiving an offer from remote participant user agent with the *qos-dds-send* attributes (in the Invite message), the proxy SIP forwards them to the COPS-PEP for translation into network QoS query to reserve network resources in the edge router. In the receive direction, those attributes correspond to the QoS query a participant uses in a *qos-dds-recv* attributes in the answer. When a participant receives an offer with *qos-dds-recv* attributes, the proxy SIP uses the corresponding QoS translation mechanisms it supports in the send direction, and then includes them in a *qos-dds-send* attributes.

In all the cases described above—and once the offer/answer exchange completes—both sender and receiver use the attributes embedded in the SDP messages to perform resource reservation. If a participant does not succeed in using the mechanism in *qos-dds* attributes, it should use default QoS settings defined in [14]. If a participant unsuccessfully tries all the common QoS mechanisms in the *qos-dds* attributes for a given direction, there may be network entities (such as Proxy SIP entities) that cannot process the message.

2.7 Integrating All the Pieces Together

Having described the architectural pieces of our solution above, we now describe how we integrated these pieces and show the interactions among the entities that help realize cloud-based DDS applications. The specific DDS–SIP/SDP interactions that take place in our solution are summarized below and illustrated in Figure 3.

- DDS entities interested in publishing data specify the DDS media description by sending a SIP control message to the core network. If the network core accepts the query for resources the required resources are allocated, at which point the application can securely and reliably exchange data. When a call setup is initiated, the caller application calls the SIP session setup through the proxy SIP. The proxy SIP encountering the caller message starts the QoS session to interact with the remote Proxy SIP and the QoS mechanisms in the edge router.

- When the host application specifies or modifies its QoS requirements, it sends an INVITE message to the proxy SIP which intercepts it and redirects to the destination for notification (offerresponse contract). Subsequently, the receiver node adapts its DDS QoS policies with those notifications and sends a response to its proxy SIP, which notifies the COPS Policy Enforcement Point (PEP) with the new QoS requirements.
- In the discovery phase, DDS entities (DomainParticipant, DataWriters and DataReaders) in distributed nodes find out about each other using the default EndPoint Simple Discovery Protocol (ESDP) [9]. During this process the DomainParticipant details are communicated automatically to all other DomainParticipants in the same Domain by sending/receiving discovery messages—also known as participant DATA messages. The discovery phase integrates the SIP service to enable the communication between SIP and non SIP applications, that is, it assess the interoperability between distributed applications that are intended to share data with DDS applications.
- The transport phase actually sends and receives messages over the underlying network. This phase integrates the network QoS services, *e.g.*, resource reservations.

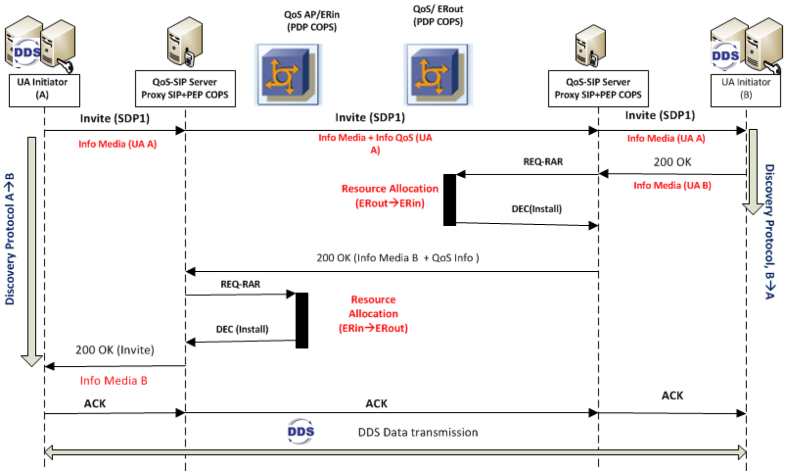

Fig. 3. Architecture of the Signaling Flow Description

3 Realizing Cloud-based DDS on DiffServ IP Networks

This section describes a concrete realization of our cloud-based DDS solution using SIP/SDP for DiffServ IP networks.

3.1 Implementation Overview

Below we describe how we implemented key elements of our architecture shown in Figure 1.
SIP proxy. In our implementation, we modified the JAIN SIP Presence Proxy [6] to support the DDS QoS policies as described by the SIP standard: we added those policies as new attributes in the Jain SIP Stack [7], and modified the proxy to support the communication with COPS-DRA server (explained below) which we created for this purpose. In fact, the QoS requests are handled by the edge router, *Juniper Mi7 in our case*, which implements all mechanisms to perform the admission control decision with the help of the COPS-DRA server.
COPS-DRA. The COPS-DRA is used on the interface between the edge router and the Bandwidth Broker (BB). The BB is the COPS-PDP entity in charge of controlling of provisioning resources for the network. It offers the capability of provisioning resources to the local DDS client and make easy to request the update of allocated resources.
Edge router. We used Juniper M7i, which is a multi-service edge router that incorporates network backbones needing reliable, secure and high-performance IP WAN connectivity, Internet access and services. It delivers rich Layer 3 services including granular per logical interface QoS, hardware-based IPv6, and multicast.

To describe the new offer/answer extension to support signaling for DiffServ infrastructure, we used the attributes shown in Table 4, which are defined in [8] and developed them using the same JAIN SIP Stack.

Table 4. SDP Attribute Values for Cloud-based DDS QoS in DiffServ Networks

```
attribute =/ qos-dds-send-attr
attribute =/ qos-dds-recv-attr
qos-dds-send-attr=’qos-dds-send’ ’:’[[SP]qos-dds*(qos-dds))]
qos-dds-recv-attr=’qos-dds-send’ ’:’[[SP]qos-dds*(qos-dds))]
qos-dds = ’0 20’ ’0 10’ ’R’ ’12’
qos-dds = ’0 0’ ’0 50’ ’U’ ’20’
qos-dds = ’0 0’ ’0 0’ ’U’ ’46’
```

3.2 Signaling Procedure

We now describe how the DDS and SIP/SDP mechanism are integrated in the context of DiffServ networks and demonstrate the QoS provisioning capabilities provided with the help of the COPS-DRA bandwidth broker (called also COPS Server) for pub/sub DDS sessions in cloud environments.

Contemporary DDS implementations contain a set of commonly used transport plug-ins (such as UDPv4 and UDPv6), as well as extensible transports (such as Secure WAN Transport). The DDS transport priority QoS describes the DSCP Field used by the DiffServ edge router classification. Packet classification features provide the capability to partition the DDS traffic into multiple priority levels. During packet classification, the DDS client performs a lookup and assigns a QoS label to the packet. The QoS label identifies all QoS actions to be performed on the packet and from which queue the packet is sent. The QoS label is based on the DSCP value in the packet and decides the queuing and scheduling actions to perform on the packet.

In our architecture, the DDS transport priority QoS policy is encoded within SIP/SDP messages, and partitioned into 4 Classes of Service (CoS) in the context of COPS-DRA-based network: Real Time (RT), Non Real Time 1 (NRT-1), Non Real Time 2 (NRT-2) and Standard (STD) as shown in Table 5.

Table 5. DDS Class of Service in the DSCP Field

PHB	CoS	dropping probability	DSCP	
			Binary	Decimal
EF	RT		101110	46
AF1	NRT1	Low	010100	20
AF2	NRT2	Medium	001100	12
BE	STD	High	000000	0

Each edge router applies the following DiffServ concept: packets belonging to real-time streams are handled in the router according to the same Per Hop Behavior (PHB) defined for the RT CoS, and packets belonging to near real-time streaming and High Throughput Data are handled in the router according to the PHBs defined for the NRT-1 CoS and NRT-2 CoS, respectively.

QoS policies mapped into SIP messages are used by the Proxy SIP. When the host changes its QoS requirements, therefore, the SIP message is intercepted by the proxy SIP to be redirected to the destination for notification (offerresponse contract). The receiver node then adapts its DDS_QoS policies with those notifications and sends a response back to its proxy, which notifies the PEP with the new QoS requirements to adapt its class of Service (CoS).

3.3 Session Management and QoS Setup for DDS on the Cloud Network

After the SIP UA is configured to make and receive calls, the processing starts with the registration phase in the “Registrar” (user registration within the SIP location database) to make all users accessible from all other SIP Servers: the

Proxy SIP delivers SIP REGISTER request from UA to the "Registrar" to check the authentication, authorization and accounting using the authentication headers [16]. The scenario shown in Figure 4 presents a detailed description of the signaling procedure and all mechanisms included in the QoS reservation.

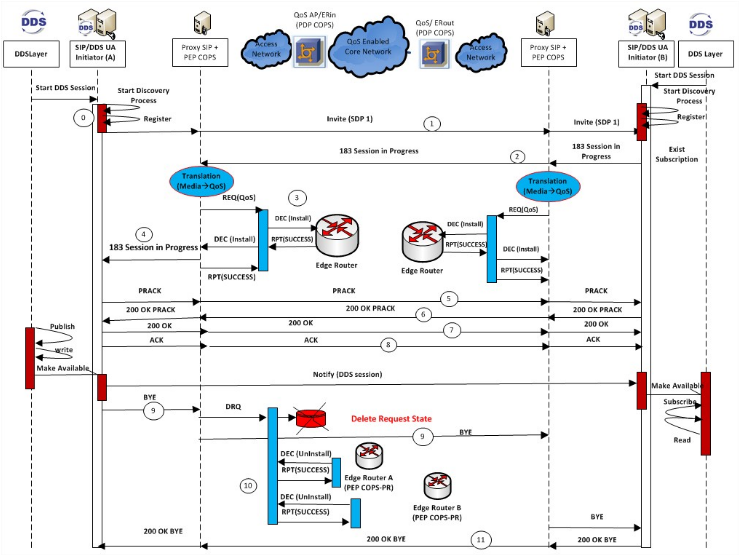

Fig. 4. QoS Support and Session Establishment for DiffServ

The signaling procedure shown in Figure 4 comprises the exchange of messages from DDS-SIP (*i.e.*, Start DDS session, Invite, 183 Progress, Prack, 200 OK Prack, 200 OK and Ack) and the messages from COPS protocols (*i.e.*, Decision, Report, Reserve and Response). This procedure is started when a caller (user A) sends a standard SIP *Invite* message and is completed upon receiving a corresponding standard SIP *ACK* message. The *200 OK* message that reaches the callee (user B) already informs it about successful connection establishment. Accordingly, the setup duration is given by the time elapsed between the moment of sending Invite and receiving *200 OK* message.

The following example shown in Table 6 describes an IP communication session described with SDP. This corresponds to a classical usage of SDP message to convey real-time media component including audio and video transported over RTP and including the PCM codec used for the audio (payload type=0), and also a media line for video with the H.261 codec (payload type=31). The bandwidth requirements are included in line 'b', the QoS requirements in the sending direction are included within the 'qos-dds-send', that is, the application requires low latency, minimum deadline, reliable transport (reliability in DDS means using of NAck requests of the reliable multicast protocol) and its required PHB behavior (AF) is fixed to 12. With respect to Table 5, this application has medium dropping probability. The DiffServ mechanisms in the edge router (classification, policing, shaping, dropping) take those parameters from the COPS-DRA server and process packets with respect to this query.

Table 6. Body SDP Carried Within the SIP Invite Message

```
v = 0
o = alice 2890844526 2890844526 IN IP4 host.ocean.com
s = -
c = IN IP4 ahkiri.laas.fr
t = 0 0
m = audio 49170 RTP/AVP 0
a = rtpmap:0 PCMU/8000
m = video 51372 RTP/AVP 31
a = rtpmap:31 H261/90000
b = AS:512
a=qos-dds-send: 0 0 0 0 R 12
a=qos-dds-recv: 0 50 0 5 R 12
```

Session setup. Participants exchange `INVITE` and `Session in Progress` messages to negotiate the media description and the QoS setting that the session should fulfill: An INVITE message is sent with all information about the participant (IP address, port number, QoS attributes, CoS description, media description) to establish the session between the publisher and the subscriber. The Proxy SIP "A" then adds the address of record to attract all other messages to pass through. Since the destination address is found in the local database, the message will be redirected to the destination, precisely to the remote SIP UA which starts the `Session in Progress` message with its media description. After the session description attributes, *e.g.*, QoS attributes, are known, the QoS negotiation for resource reservation begins.

The *Invite* message carries the caller URI within the SIP header and the session specification within the body SDP (session description, media description, source port, codec, DDS-QoS) as shown in Table 6. The proxy SIP intercepts the invite message, and on the basis of the information given in the body SDP decides whether to start a QoS session or not. If so, it inserts the required descriptors within the Invite message and forwards it to the callee.

The message can also be intercepted by any other SIP server encountered in its destination or any QoS-aware SIP proxy (since DDS QoS attributes are not recognized by classical SIP server, they are just ignored when it processes the Invite message). Hence, the Proxy SIP is present during all steps of SIP-based communication described below, including the registration (to access to the registrar server matching two SIP-UA requests) and the session establishment (forcing all SIP and SDP messages to pass across it with the header "record route").

Since the proxy SIP has all required information to request the QoS reservation to the edge router for the SIP UA (B), it sends a request to the COPS-DRA server (Bandwidth Broker or BB) to translate the media description fields into a QoS specification understandable by the edge router BB is in charge of automating the QoS resource reservation process. It sends QoS requests to the router to install them (DEC or Decision, RPT or Report, REQ or Request).

At the same time, the QoS information is sent within the SIP `183 Session Progress` message to the other proxy SIP at the border of the access network of the SIP UA (A). Similarly, this information is stored by the SIP Proxy to maintain a trace of the current session, called `QoS State`. Consequently, it send a QoS request to the COPS-PDP in its domain to claim the QoS reservation at the edge of the network. The PDP connected to the edge router has to automate the QoS reservation process based on the QoS information it has received from the proxy SIP.

At this point, the caller UA (A) sends a Prack (Reliability of Provisional Responses in SIP) message to confirm the `183 Session Progress` message has been received and waits for the `200 Ok Prack` message sent by the caller UA (B) to notify the session setup. Moreover, the caller proxy SIP inserts the `Caller ER` filed into its message including the IP address of the caller endpoint. The caller proxy SIP uses this field to specify the remote address of the endpoint where the reservation request have to be send to QoS provider.

The caller proxy SIP adds its VIA field to the message intercepted from the sender UA (A) in which it adds some specific information (IP address, source port, caller address) to maintain trace of the QoS State. Finally, the session establishment is confirmed when both caller and caller Proxy SIP exchange `200 OK` and `ACK` SIP messages. The sender application performs the data delivery to the remote participants, since the signaling path is established as described herein, the data path is taken by the data packets according this signaling process.

QoS negotiation. The RT CoS covers the requirements to offer strict QoS assurance of low packet delay, low packet losses, and low jitter. As a consequence, we focus on admission control methods that support such assurance. The NRT-1 CoS covers the requirements to offer strict QoS assurance of low packet losses and medium packet delay. In NRT-2 CoS for inter-domain link, we follow the QoS requirements related to the High Throughput Data end-to-end CoS that are expressed by the assurance of minimum throughput R_{min}. We do not consider any QoS assurance support for STD CoS, though an amount of inter-domain link capacity and buffer must be dedicated to this CoS, as well.

Moreover, since DDS is based on Offer/Request contract between distributed entities, this model can be used to enhance adaptivity and robustness of communication. For example, consider user agent A in Figure 4 that is responsible for the resource allocation. In this case Proxy A follows the “183 session progress” and performs the translation of the “qos-dds” attributes included within the media information to QoS requirements.

The “qos-dds” attributes include the *DiffServ Code Point* (DSCP) priority that the DDS application adds to SIP/SDP signaling messages. The PEP uses this value for traffic classification in each border router. Moreover, QoS settings (such as latency budget, deadline in the "a" line in Table 3) are translated into bandwidth, latency, and jitter. The PEP then sends the "REQ" message to the PDP in its domain to perform the resource reservation using the message context with the appropriate identification and authorization of the participant. As a result, the PDP analyzes the request (consults its database for verification), performs the DiffServ resource allocation using COPS Provisioning mechanisms, and sends a decision "DEC" to the PEP including the DSCP value for the media stream which generates a report message "RPT" to indicate the success of the decision.

The "*183 session progress*" message is also sent to proxy A to indicate successful resource allocation in its direction. The DDS discovery protocol performs at both direction endpoints discovery from the information given by its near proxy and registrar to connect different DDS domains. The rest of the process is performed as described above and the DDS streams are exchanged between participants. After a pair of remote participants have discovered each other, they can move on to the Endpoint Discovery phase, which is how DataWriters and DataReaders find each other. During this phase DDS matches DataWriters and DataReaders by sending publication/subscription declarations in DATA messages including information (Globally Unique ID or GIU, QoS) about the application.

Terminating the session and releasing the resources. DiffServ entities should release the allocated resources after the session termination. The proxy SIP therefore performs the resource liberation from the PEP and the PDP. In particular, the "uninstall" message is sent to perform this operation and a negotiation process executed by PDP and PEP.

4 Related Work

This section compares our work on DDS Middleware-based QoS provisioning with related research on pub/sub capabilities over WANs, which are an important part of cloud environments. We therefore focus on related work comprising pub/sub efforts for WANs.

QoS management in content-based pub/sub middleware: QoS management in content-based pub/sub middleware includes the *Publish/Subscribe Applied to Distributed Resource Scheduling (PADRES)* [15]. PADRES allows powerful content-based routing mechanism based on the message content instead of IP-based routing. For example, a mechanism for congestion avoidance to optimize the network resource usage in Content-Based Routing Networks is presented in [4]. Despite this advantage, in practice core network resource allocation algorithms need to know more about an application and how to process its QoS requirements within the core routers.

In our approach, DDS provides the DCPS services described in Section 2.1. These services make it possible to pub/sub-structured data as a portion of distributed relational information model. Moreover, our approach focuses on how to use existing middleware and mapping it to provide the QoS for existing applications.

Network QoS broker in middleware: Related work in this area focuses on integrating the signaling process into the QoS provisioning mechanisms. For example, message-based signaling middleware for the control plane to offer per-class QoS is described in [21]. In practice, however, it is insufficient to deploy routers to assure the QoS between two hosts at the control plane. Likewise, a network communication broker to provide per-class QoS for multimedia collaborative applications is presented in [5]. This work, however, supports neither mobility service management nor scalability since it adds an interface to the application and middleware for QoS notification when an event occurs in the network and the application should adapt to this modification.

In contrast, our approach uses a SIP framework to manage sessions in the service plane. This framework provides interoperability and network QoS negotiation with the help of COPS-DRA Bandwidth Broker for adaptability and robustness. Our solution therefore requires no modifications to applications.

Network QoS management: Most related work on DRE system QoS support focuses on solving QoS-constrained routing problems separately from the middleware QoS policies. There have been several efforts, however, to bring SIP protocol into the existing IP architecture. Current research includes adaptive and reflective middleware, and middleware for mobile and ubiquitous systems. For example, a programmable networking approach to provide QoS as component-based architecture is described in [3]. The authors in [13] designed a basic DDS/SIP Gateway to connect remote DDS domains, though they do not focus on the core network behavior and how resources are managed. The Common Open Policy Service (COPS) was extended in [11] to support SIP signaling over wide area networks, but COPS/SIP has not yet been integrated with DDS.

Despite the promise held by the COPS extensions, it does not account for the application needs of pub/sub DRE systems. Moreover, there is no straightforward approach to integrate SIP/SDP with DDS. Similarly, [20] is a mobility service proxy that deals with media session management for DDS mobile nodes.

Consequently, to bridge the gap between DDS and SIP/SDP so that the desired QoS policies are supported both by the end-system middleware and the network, we defined a new SIP Signaling Class of Service (S-CoS) for transferring signaling messages. Our approach does not underutilize the resources because it reserves only the required resources, which performs packet classification, policing, and traffic shaping in the edge router. For RT CoS and NRT CoS, our approach does not use all the resources since 80% of them are used by Best Effort traffic.

Table 7 compares the related work presented in this Section with our approach. R1 refers to the QoS management in content-based publish subscribe middleware and R2 refers to the Network QoS broker in middleware. The adaptive and reflective approaches in R1 and R2 work well today when they receive all the resources required. Other research has shown, however, that over provisioning resources has several drawbacks in scalability and can fail completely under the slightest anomaly [19]. Our approach therefore provides an adaptive framework that enables the end-to-end QoS provisioning over multi-domains independently of the underlying transports.

Table 7. Comparing the Various Approaches

Features	R1	R2	Our approach
Content-based pub/sub middleware	*		*
QoS specification	*	*	*
Resource allocation		*	*
QoS assurance			*

5 Concluding Remarks

Despite the powerful QoS mechanisms provided by pub/sub middleware, such as OMG's *Data Distribution Service* (DDS), these technologies are currently confined to end-systems, so they can only control local resources in DRE systems. To support end-to-end QoS in clouds, therefore, pub/sub DRE systems require an approach that enables resource reservation over wide area networks, yet is transparent to application logic. This paper addresses these requirements and describe a solution that seamlessly integrates DDS with SIP/SDP over clouds as follows:

- At the service plane, DDS applications use SIP signaling messages that allow senders to contact receivers to obtain their IP addresses and to agree the media description and "qos-dds" attributes.
- At the control plane, the network QoS provisioning mechanism provided by the COPS-DRA entities encodes application QoS requirements embedded within the SDP messages supplied to the network elements. The COPS protocol is chosen as common signaling mechanism to enforce the policy control mechanisms (*e.g.*, QoS negotiation, coordinate the data path and signaling path management, performs resource reservation, etc.).

We use DDS latency, transport priority, and bandwidth QoS settings to map them to signaling class of service (S-CoS) that allows DDS applications to control the data delivery in the access and core network for resource provisioning. To help preserve compatibility with existing DDS applications, we impose minimum requirements on the set of QoS policies that are compatible with most deployed DRE systems. Our future work will focus on extending the set of supported DDS QoS policies using the *Next Steps In Signaling* (NSIS) protocol and extensive empirical measurements to validate our approach.

References

1. K. Birman, G. Chockler, and R. van Renesse, *Toward a Cloud Computing Research Agenda,* SIGACT News, Volume 40, Number 2, pp. 68–80, 2009.

2. Camarillo, G., Ed., Marshall, W., Ed., and J. Rosenberg, "*Integration of Resource Management and Session Initiation Protocol (SIP)*", RFC 3312, October 2002.
3. L. Capra et al.,'*Reflective Middleware Solutions for Context-Aware Applications*', Proceedings of the Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, pp 126 - 133, 2001
4. M. Chen, S. Hu, V. Muthusamy, and H.A. Jacobsen. Congestion Avoidance with Selective Filter Aggregation in Content-Based Routing Networks. Middleware Systems Research Group, Nov 2010.
5. Chi, Z., Sadjadi, M., Weixiang, S., Raju, R., and Yi, D.,'*A user-centric network communication broker for multimedia collaborative computing*', International Conference on Collaborative Computing: Networking, Applications and Worksharing, 2006. CollaborateCom 2006.7-20 Nov. 2006, pp1-5
6. JAIN-SIP-PRESENCE-PROXY, `snad.ncsl.nist.gov/proj/iptel/nist-sip-downloads.html`
7. NIST-SIP, `snad.ncsl.nist.gov/proj/iptel/`
8. E-H. Cho, K-S. Shin, S-J. Yoo, *SIP-based Qos support architecture and session management in a combined IntServ and DiffServ networks*, Journal of Computer Communications, Volume 29 Issue 15, September, 2006
9. *Data Distribution Service Interoperability Wire-Protocol Specification.* DDSI v2.1, `www.omg.org/spec/DDSI/2.1/`
10. D. Durham, Ed. et al, "*The COPS (Common Open Policy Service) Protocol Status of*", RFC2748, January 2000.
11. G. Gross et al, *COPS Usage for SIP*, draft-gross-cops-sip-01.txt, IETF Draft.
12. Handley, M., Jacobson, V., and C. Perkins, "*SDP: Session Description Protocol*", RFC 4566, July 2006.
13. J.M. Lpez et al, *DDS/SIP Interworking: A DDS-SIP Gateway*, OMG Workshop on Real-time, Embedded and Enterprise-Scale Time-Critical Systems. May 24-26, 2010, Westin Arlington Gateway, Arlington, VA USA.
14. OMG-DDS, "*Data Distribution Service for Real-Time Systems Specification*". DDSv1.2, `www.omg.org/spec/DDS/1.2/`
15. PADRES, `padres.msrg.toronto.edu/Padres/WebHome`
16. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "*SIP: Session Initiation Protocol*", RFC 3261, June 2002.*
17. Rosenberg, J. and H. Schulzrinne, "*An Offer/Answer Model with Session Description Protocol (SDP)*", RFC 3264, June 2002.
18. D.C. Schmidt et al, *Middleware R&D Challenges for Distributed Real-time and Embedded Systems*, ACM SIGBED, Volume 1 Issue 1, April 2004.
19. Yaqing Huang and Roch Guerin, *Does Over-Provisioning Become More or Less Efficient as Networks Grow Larger?*, Proceedings of the 13TH IEEE International Conference on Network Protocols, 2005, pp. 225-235.
20. Ki-Jeong Kwon, Choong-Bum Park, Hoon Choi, *A Proxy-based Approach for Mobility Support in the DDS System*, 6th IEEE International Conference on Industrial Informatics, 2008. INDIN 2008.
21. Teodora, G., et al., '*A Session Initiation Protocol based Middleware for Multi-Application Management*'. IEEE International Conference on Communications - ICC, Multimedia Communications & Home Services Symposium (Glassgow, UK, 2427 06 2007).