QOS-AWARE MULTIMEDIA WEB SERVICES ARCHITECTURE

Ikbal Taleb, Abdelhakim Hafid
[1]*Network Research Laboratory, University of Montréal, Pavillon André-Aisenstadt H3C 3J7, Canada*

Mohamed Adel Serhani
[2]*Concordia University, 1455 Maisonneuve Blvd West, Montreal, Quebec, H3G 1M8, Canada*

Keywords: Web Services, End-to-End QoS, Web Services QoS Broker, Network Resource Management.

Abstract: Due to the increasing growth of Web Services, Quality of Service (QoS) is becoming a key issue in web services community. Providers and clients need to use QoS-aware architectures to get/ensure end-to-end QoS. The QoS delivery to clients is highly affected by the web service performance itself, by the hosting platform (e.g., Application Server) and by the underlying network (e.g., Internet). Thus, even if web services together with hosting platform provide acceptable QoS, they also require sufficient available network resources to deliver end-to-end QoS. In this paper, we propose a solution approach to the problem of end-to-end QoS support for web services. Our approach rely on the utilization of a web service, called Network Resources Manager (NRM), to take care of the QoS support in the network connecting the client host and the matching web service location. NRM either relies on the network QoS capabilities (e.g., Integrated Services, Differentiated Services, Multiprotocol Label Switching), if any, or uses a measurement-based scheme to estimate the quality that can be delivered between the two locations. One of the key differentiator of our solution is that it does not require any changes to the currently used infrastructure by the users and web services providers.

1 INTRODUCTION

In Service Oriented Architectures (SOA), both service providers and service users should be able to define QoS related statements. This is needed to enable QoS-aware service publication, discovery, and usage. For web services, QoS concerns the non-functional aspects of the service being provided to the users.

Estimating and guarantying QoS are important for both Web service (WS) clients and WS providers. For clients, when selecting a suitable WS prior to service usage, it is important to be informed of the QoS status. For WS providers, it is a competitive edge over others that provide the same web services without QoS support.

Multimedia Web Services present additional challenges that are different from those of traditional Web services. QoS of a streaming session depends on a combination of factors, ranging from the characteristics of the streaming sources (e.g., link capacity, availability, and offered rate) to the characteristics of the network paths (e.g., available bandwidth, packet loss rate, etc.).

The design of Network Resources Manager (NRM) providing mechanisms, measurement strategies and network information of interest to realize high quality streaming sessions between WS Clients and WS provider is therefore a challenging task.

The proposed architecture aims at supporting end-to-end QoS at two levels (server level and the network level). For that purpose, it employs a third party broker (Adel, 2004) to assure QoS specification and monitoring at the server level, and a third party NRM to guarantee the QoS at the network level. Both components cooperate together to support end-to-end QoS between providers and their clients.

Taleb I., Hafid A. and Adel Serhani M. (2005).
QOS-AWARE MULTIMEDIA WEB SERVICES ARCHITECTURE.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 133-139
DOI: 10.5220/0001236101330139

This paper is organized as follows. Section 2 introduces web services and presents related work on QoS in the context of Web Services (WSs) including discussions on the limitations of existing approaches. Section 3, describes the NRM architecture. Section 4 presents the implementation of the proposed architecture and the simulations-based evaluation of NRM. Section 5 concludes the paper and presents future research directions.

2 BACKGROUND

2.1 Web Services

A Web service is a software system identified by a URI (Uniform Resource Identifier), whose public interfaces and bindings are defined and described using XML (eXtensible Markup Language). Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols. (Web Service Architecture, 2003)

A web service is invoked from any application; but executed in the remote host server. Web services usually use Hypertext Transfer Protocol (HTTP) as a fundamental communication protocol which carries exchanged SOAP messages between clients and web services.

Web Services (WSs) provide a new architecture paradigm for building distributed computing applications based on XML. The Web service functionalities are exposed through an interface description and are publicly available for use by other programs. Web services make use of standard Internet protocols, such as: **SOAP** (Simple Object Access Protocol) which is an XML based protocol for messaging and remote procedure calls. **WSDL** (Web Services Description Language) which is a formalized XML based language for describing web services. **UDDI** (Universal Description, Discovery and Integration) which is specification for publishing and discovering web services description through public registries.

Web Services Architecture, is based on the interactions between three roles: service provider, service registry, and service requester. The interactions involve publish, find, and bind (interact) operations.

2.2 Related Work

Research on web services has focused more on functional and interfacing issues, i.e., Simple Object Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). Recently, QoS issues began receiving more attention from the web services community. QoS is not new to distributed computing systems community, but in web services there are new issues related to web services properties. For web services, QoS have to include network properties according to the public network (i.e., Internet). Clients are using Internet to invoke web services; currently, the Internet treats all traffic equally as 'best effort' and provides no support for QoS.

A sizeable amount of research on Web services QoS concern semantic definition of web services and QoS constraints. DAML-S (DAML-S, 2002) supports semantic description of web services, including specification of functionalities and QoS statements. IBM introduced (Keller, 2002) WSLA (Web Service Level Agreements), which is an XML specification of SLA (Service Level Agreement) for Web Services, focusing on QoS constraints. A Carleton University group (Tosic, 2003) developed the Web Service Offerings Language (WSOL) for the formal specification of various constraints, management statements, and classes of service for Web Services. None of those address the problem of providing end-to-end QoS when a web service, that satisfies the user QoS requirements, is invoked.

A. Shaikh Ali and al. (Ali, 2003) from Cardiff University propose UDDIe, as a new registry and an extension to the UDDI standard. UDDIe supports the recording of user defined properties associated with a web service, and to enable its discovery based on these properties. This work extends the UDDI to integrate QoS descriptions and search-operations capabilities. However, when a web service, that satisfies the user QoS requirements, is selected, there is no guarantee that the network will support the requested QoS. For example, if the published audio quality of a web service (e.g., music player) is "CD quality" and the user requires audio quality of "CD quality", the web service will be selected as satisfying the user requirements; however, the user will get this quality only if the network has enough availabe resources to provide this quality.

M. Tian et al. (Tian, 2003) introduce a scheme of QoS integration in web services. It is based on an XML schema for Web Services QoS definition; it includes mechanisms for efficient selection of QoS-aware web services, support of dynamic QoS mapping at runtime, and support of instant QoS information delivery. They introduce an entity, called QoS proxy, which is located between the transport layer and the web service layer. Its role is to mark outgoing packets in the case of DiffServ (Blake, 1998) enabled network. It can be also used

with other networking technologies, such ATM and UMTS. The main drawback of this approach is that it requires changes to the protocol stack in all involved entities (i.e., users and providers). This is in addition to security concerns; indeed, malicious users can mark their outgoing packets to get the best service available (e.g., Expedited forwarding).

In this paper we present an architecture that allows the service broker to select web services that can be delivered while satisfying the user end-to-end QoS requirements. Our proposed architecture does not require any changes to the protocol stack of involved entities. Indeed, our proposed approach is based on a web service, called Network Resource Manager (NRM), that is responsible of ensuring that the network (between the selected web service and the user) supports its part of the required end-to-end QoS. NRM is invoked only when the web service provider provides its part of the required QoS (e.g., in terms of CPU of the server executing the web service). Thus, our proposed architecture extends exiting approaches (e.g., UDDIe) to support end-to-end QoS. For example, when UDDIe returns one or more web services that satisfy the user requirements including QoS requirements, NRM can be used to identify the web service, if any, that when invoked will satisfy the user end-to-end QoS requirements. The Network Resources Manager (NRM) is a web service capable of measuring and checking end to end QoS properties that helps in selecting suitable web services.

3 NRM WEB SERVICE

The Network Resource Manager (NRM) is a fundamental addition to the WS QoS-based broker Architecture (Adel, 2004); it is involved in a number of the broker's tasks. It will play a major role in delivering end-to-end QoS guarantees in SOA.

As a Web Service, the NRM publishes its interface description in the UDDI/UDDIe registry to respect the Service Oriented Architecture. Once available via registries, interested components can invoke their operations.

The NRM WS performs a number of key operations that are necessary in supporting the operations of the QoS broker. Its main objective is the support of QoS in the network that is used to deliver the requested web service from the provider's web service hosting platform to the user's location.

The key task of NRM is to assist the WS QoS Broker in the web services selection process in response to a user request. Indeed, when the QoS broker identifies list of web services that satisfy the user requirements, it invokes the NRM to check the capability of the network, between the web service location and the user location, to support he required QoS.

NRM implements the SOAP handler class to intercept messages coming from the broker (Figure 1). Upon receiving an invocation request, the handler forwards the request to the NRM analyzer that parses the request and extracts information of interest, such as QoS parameters and their values. Then, the analyzer sends the extracted information to the NRM Mapper that performs mapping of QoS parameters provided by the broker and QoS parameters of the network. The Mapper provides the NRM checker with the QoS parameters to be supported by the network.

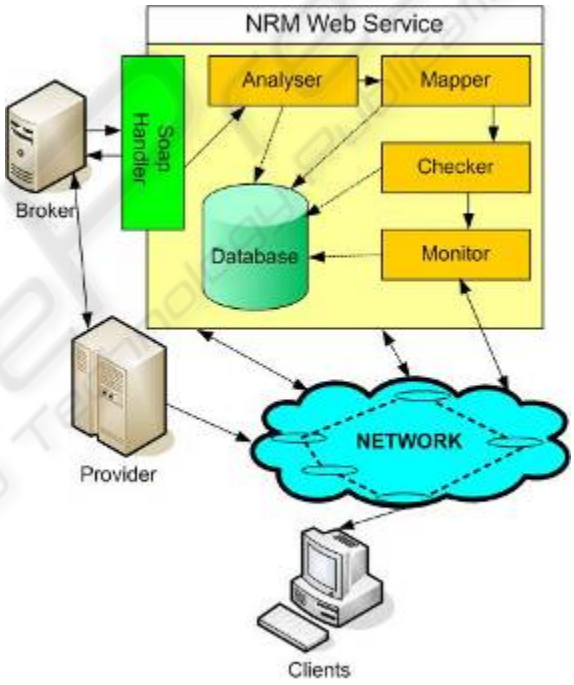

Figure 1: Broker and NRM interactions

NRM (or rather the NRM checker) uses a number of mechanisms to realize this task. If the underlying network supports QoS, such integrated services (IntServ (Braden, 1994)) or differentiated services (DiffServ (Blake, 1998)), then NRM uses these capabilities to support QoS. For example, in the case of IntServ, it uses RSVP (Braden, 1994) (Resource Reservation Protocol) to make the necessary resources reservation to meet the required QoS. In the case of DiffServ enabled network, it marks outgoing packets according to the required QoS (e.g., use Expedited Forwarding (EF) marking to support voice/video) to provide differentiated

services or it can use the services of a bandwidth broker (Stattenberger, 1998), if any, to provide, if possible, the required QoS.

In cases where no such mechanisms are supported by the network, NRM (monitor in Figure1) makes use of measurement techniques to estimate the state of the network between the web service location and the user location; it uses for example probes to measure the delay and loss rate between the two locations.

NRM also performs periodic measurements between different points in the network to gather statistics about the state of the network; the results of the measurements, stored in the NRM database, can be used to estimate/predict the state of the network and thus helps in estimating the QoS that will be likely delivered between two points in the network (e.g., between a web service and a user location)

3.1 Operations Scenario

We assume that there is a number of WS Providers that have published their MWS (Multimedia Web Services) using a UDDI standard registry, or an UDDIe [4] registry to integrate QoS Data within the published WSDL. A web portal (i.e., a web based user interface) allows users to search for web services (e.g., video-on-demand). More specifically, the user enters a description of the service he/she is looking for. (*Video on demand*) including QoS requirements. The following steps are executed:

The client (Web/Java Application) searches and binds to the QoS broker web service; then, it invokes the QoS broker with the user request as an attribute (arrow 1 in Figure 2).

The BWS starts a process to retrieve a list, L, of WSs, from UDDI registries, that match the user requirements including QoS requirements (arrows 2 and 3).

The BWS considers the first web service, MWS, in L and calls the NRM web service to check whether there are sufficient available network resources to support MWS between MWS location and the user (arrow 4). Note that before invoking NRM, BWS searches and binds to NRM; NRM is just another web service.

The NRM WS analyzes the BWS Request and identifies key information it needs to process the request (arrow 5); examples are: IP address of the user, IP address of the server running MWS, and the requested QoS attributes.

If the network connecting the user and MWS is QoS-enabled, then, NRM will make use protocols/schemes provide by the network. If the network is IntServ-enabled, thus, NRM will use RSVP to make resources reservation between the user and MWS. If the network is DiffServ-enabled, then NRM will mark the outgoing packets (e.g., video traffic generated by MWS) according to the requested QoS or will make use of the network bandwidth brokers if they are available. If the network is not QoS aware, then NRM use probes to check the status of the network (or rather network path) between the user and MWS. In the prototype implementation of NRM (see Section 5), NRM marks outgoing packets assuming that the network is DiffServ enabled.

If NRM is successful in reserving the required resources or estimates (in the case the network is not QoS aware) that there are enough resources to support the request, it returns an accept response to BWS; otherwise, it returns a rejection (arrow 6).

If BWS receives a rejection, it considers the next web service in the list L and calls NRM and repeats the same process (arrows 4 and 5). This process ends when an accept response is received or when all web services in the list L are considered without any success. In the case of an accept, MWS is returned to the user; in he other case, a rejection is set to the user

If the user receives a rejection, then it gives up or changes his/her request in terms of QoS requirements (i.e., starts a renegotiation); otherwise, he/she uses the WSDL document provided by the Broker (arrow 7) to bind to MWS and invokes the service (arrow 8). Then, the provider server starts media streaming towards the client using RTP protocol (arrow 9).

NRM can be as complex and/or as sophisticated as the NRM providers want. For example, NRM can support advance reservation (Hafid, 2005); in this case, in response to BWS invocation, NRM checks resources availability across all the involved networks, computes, and returns to BWS the QoS that can be supported for the time the request is made (i.e., immediately), and at certain later times carefully chosen. As an example, if the requested QoS cannot be supported for the time the service request is made, the proposed approach allows to compute the earliest time, when the user can start the service with the desired QoS.

4 IMPLEMENTATION

We implemented a Multimedia web service (i.e., video-on-demand) that provides users with the functionality of selecting and playing a movie. Movies and their metadata are stored in a MySql local database. We implemented in MWS several functions to access data and to send and receive Media contents.

We also implemented NRM as a web service that uses DiffServ marking (Blake, 1998) to provide acceptable video quality. The underlying network, connecting users and our MWS instances is DiffServ enabled.

In our prototype implementation, we used Bea Weblogic (Bea WebLogic, 2004) Workshop to design web services with a J2EE compliant environment. The Weblogic Server is the main platform deploying and publishing web services. The Java Media Framework API is included to send Video-Audio content via the network (internet). JMF (JMF, 2004) uses RTP protocol (UDP based) to transmit and receive data.

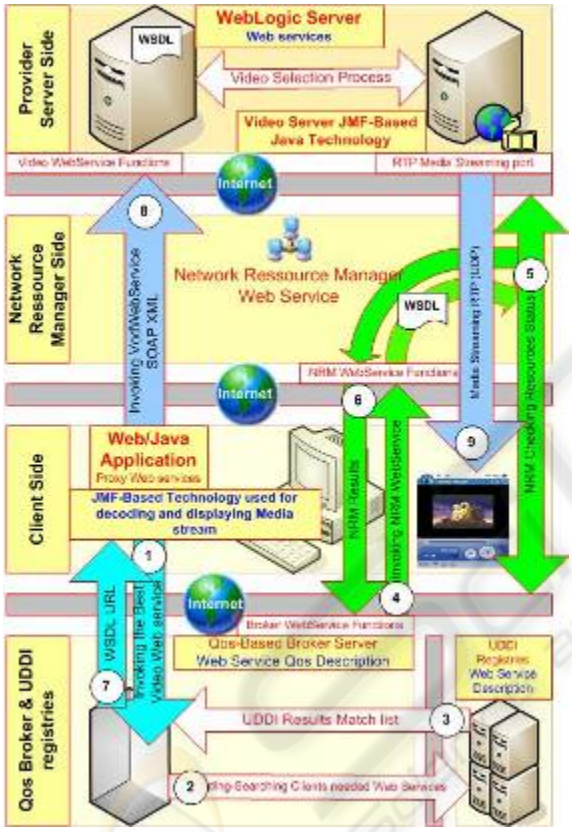

Figure 2: Architecture Components

The user is provided with different classes of QoS, he/she can select when requesting a movie; each class of QoS is characterized by the cost of the bandwidth (see Table 1). For example; a high quality video requires 5 Mbps; thus, to support high quality, the video server should have the resources to support the high quality and the network path, between the user and the video server, should allow for the transmission of 5 Mbps.

Table 1: Offered Services

QoS Classes	Video Quality	Cost	Bandwidth needed
Class 1	High	5$	5mbps
Class 2	Good	2$	2mbps
Class 3	low	1$	100 kbps

To check whether the WS provider is able to support a user request, we assume the existence of a table that specifies the maximum numbers of users that can be supported given a QoS class (the capacity of a service to support a given number of clients can be easily measured; thus, we can drop the table assumption). It also keeps track of the number of users using the service. With this information, one can easily infer whether a new request can be supported or not by the video server. For the network QoS support, NRM marks video packet with Expedited Forwarding (EF) marking to assure better QoS for video traffic.

Table 2: Example of Video Server availability

QoS Classes	Video Quality	Available ports	Max ports	Used ports
Class 1	High	2	5	3
Class 2	Good	14	15	1
Class 3	low	10	25	15
Total		26	45	19

Figure 2 depicts the interactions between the different components of the prototype implementation. Numbered arrows represent the order of interactions in time. Each arrow includes a description of the corresponding interactions; for example, arrow (1) depicts the invocation of the web service broker by the client while arrow (9) depicts the video streaming using RTP.

Figure 3 shows the multimedia web service interface to the client; indeed, when the client binds to the selected web service (in this case a video-on-demand service), it is provided with graphical user interface to select the movie he/she wants to play.

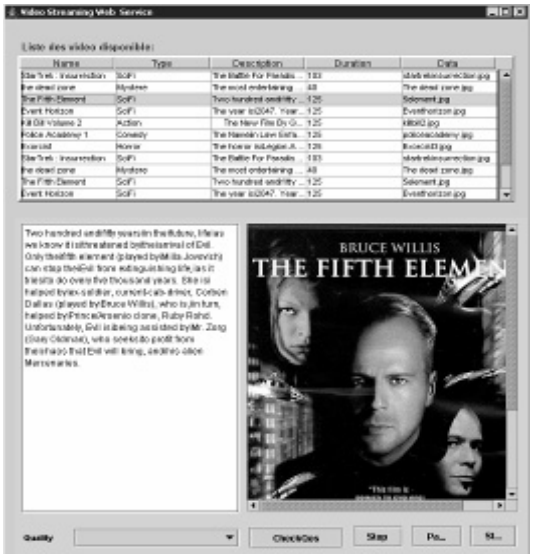

Figure 3: VoD Web Service Client

4.1 Simulations

We run simulations to evaluate the video quality delivered to the user using our prototype with NRM and without NRM support; the objective is to show that when NRM is used the user is assured to get better QoS. In fact, when the network is overloaded with traffic of other applications (such as FTP), if NRM is not used, the user receives degraded QoS (the video traffic is not marked accordingly). In the case, NRM is used, the user receives the requested QoS; in this case, NRM marks the video packet with EF marking.

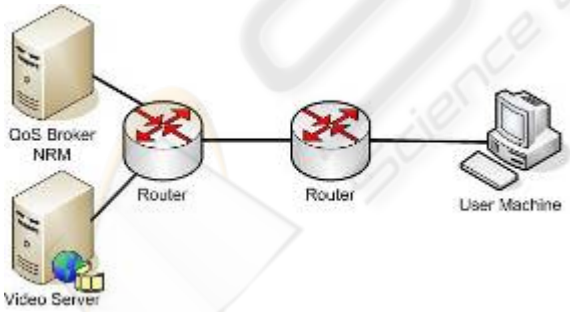

Figure 4: Testbed Setup

The network test-bed makes use of a local area network, consisting of computing nodes and routing elements. Figure 4 depicts the network setup, with the computing nodes representing the user, the video server, and the QoS Broker with the NRM web service. For the routing elements we used the `iproute2` package in the Linux operating system, which has DiffServ capability. The network link between the two routers has a capacity of 0.75 Mbps.

In Figure 4, NRM uses Router capabilities to mark outgoing packets from the server, towards the user machine, with the appropriate marking (i.e., Expedited Forwarding: EF (Blake, 1998)). It does so by accessing the router, via Telnet, and performing the required configuration.

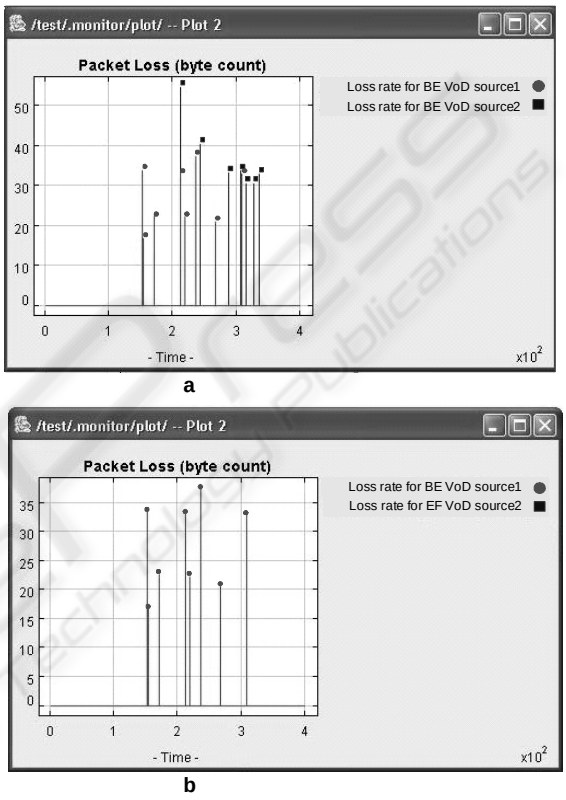

Figure 5: Loss Rate of two BE flows and EF and BE flows.

Figures 5.a shows the loss rate incured, between the video server and the user, by two video flows. Each flow is generating 0.4 Mbps of video traffic (thus, exceeding the capacity of the network link). The video flows started without involving NRM; indeed, the QoS broker selects the web service (video server in this example) based on the functionality requirements and QoS requirements of the web service provider (in this case the capacity of the video server to serve the users). The flows receive Best Effort (BE) service. Both of the flows incur losses (see Figure 5.a); thus, the QoS delivered to the user is degraded and does not meet his/her (end-to-end) QoS requirements.

Figure 5.b shows the loss rate incured between the video server and the user of two flows; each flow is generating 0.4 Mbps of traffic. The first flow

starts generating traffic without involvement of NRM; thus, it receives BE treatement by the network. The second flow starts generaing traffic with NRM involvement; indeed, NRM receives a request from QoS broker (in response to a user request to watch a movie) to check the availability of resources to deliver video traffic at 0.4 Mbps. NRM sends an accept response to start the second flow (video traffic) after configuring the router (to which the video server is connected) to mark the packets belonging to the second flow with EF. Figure 5.b shows that the second flow does incur no data losses; only the first flow incures data losses (BE flow). The reason is that the router treats EF traffic differently than BE traffic; it processes/forwads packets marked with EF first before processing/forwarding packets marked with BE.

These two simple scenarios show the need for NRM to provide end-to-end QoS for web services. Even if web services providers have the hosting platform with the capacity to provide QoS to their users, they will not succeed in satisfying end-to-end QoS without taking into account the QoS support in the network(s) connecting their hosting plaform to their users. NRM web services can be used to fill the network QoS gap that exit in today's web services deployment.

5 CONCLUSION

In this paper, we presented a solution approach to the problem of end-o-end QoS support for multimedia web services. Our solution does not require any changes to the currently available infrastructure of the users and web services providers. More specifically, we presented the design and implementation of Network Resources Manager web service. It is just another web service that is published in web services registries. It is searched, retrieved, and invoked by the web service broker. Its main role is the support, if possible, of QoS in the network connecting the matching web service location and the user location. The QoS support depends on the network capabilities in terms of QoS support (e.g., DiffServ-enabled, IntServ-enabled, and MPLS-enabled). If he network is not QoS-aware, NRM uses measurement-based approach to estimate the QoS between the two end points.

We are currently working to enhance/improve the capabilities of NRM including QoS renegotiation and advance reservation of resources.

ACKNOWLEDGMENTS

The authors would like to thank Youcef Khene, from University of Paris VI, France, for his help running the simulations.

REFERENCES

Adel, M., S., A. Hafid, Sahraoui, H. and Benharref, A., 2004. QoS broker-based architecture for Web Services. *In NOTERE.*

Ali, S. A., Omer, F. R., Rashid, A. and David, W. W., 2003. UDDIe: An Extended Registry for Web Services. *In Symposium on Applications and the Internet Workshops (SAINT'03),* Florida, *IEEE Computer Society.*

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

Braden, R., Clark, D., Shenker, S., 1994. Integrated Services in the Internet Architecture: an Overview, *RFC1633*.

BEA WebLogic platform, 2004. http://www.bea.com

DAML-S Coalition, 2002. DAML-S: Web Service Description for the Semantic Web. *In Proceeding of the International Semantic Web Conference.*

JMF, 2004. Java Media Framework API. *http://java.sun.com/products/java-media/jmf/*

Hafid, A., Maach, M., Drisi, J., 2005. DARSION: A Distributed Advance Reservation System for Interconnected Optical Networks. *In Proceedings of the 9th IEEE/IFIP Optical Network Design and Modeling (ONDM'05).*

Keller, A., and Ludwig, H., 2002. The WSLA framework: Specifying and Monitoring Service Level Agreements for Web Services. *IBM Research Report.*

Rosen E., Viswanathan, A. Callon, R., April 1999. Multiprotocol Label Switching Architecture", *draft-ietf-mpls-arch-05.txt*.

Stattenberger, G. and Braun, T., 2003. Performance of a Bandwidth Broker for DiffServ Networks, *TR, Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland.*

Tian, M., Gramm, A., Naumowicz, A., Ritter, H., Schiller, J., 2003. A Concept for QoS Integration in Web Services. *1st Web Services Quality Workshop (WQW 2003), in conjunction with (WISE 2003), Italy.*

Tosic, V., Pagurek, B., Patel, K., 2003. WSOL- A Language for the Formal Specification of Classes of Service for Web Services. *International Conference on Web Services, June 23-26 2003: LV, Nevada, USA.*

Web Services Architecture, 2003. *http://www.w3.org/*