Volume 2, No. 5, Sept-Oct 2011

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

An Overview of Enhancing Quality of Service for Enterprise Networks using Web Services

K.Manivannan*
Research Scholar
Anna University of Technology, Madurai,
Tamilnadu, India.
manivannancse@hotmail.com

C.G.Ravichandran
RVS College of Engineering & Technology
Dindigul,
Tamilnadu, India.
cg_ravi@yahoo.com

Abstract: The appearance of Web as an omnipresent media for sharing content and services has led to the rapid growth of the Internet. Content Delivery Networks (CDNs) is anticipated to provide better performance delivery of content in internet through worldwide coverage, which would be a fence for new content delivery network providers. At the same time, the number of users accessing Web-based content and services are growing exponentially. This has placed a heavy demand on Internet bandwidth and Web systems hosting content and application services. As a result, many Web sites are unable to manage this demand and offer their services in a timely manner. Content Delivery Networks (CDNs) have emerged to overcome these limitations by offering infrastructure and mechanisms to deliver content and services in a scalable manner, and enhancing users' Web experience. The planned research provides a framework designed to enhance QoS of Web service processes for real time servicing. QoS parameters of various domains can be combined to provide differentiated services, and allocating dynamically available resources in the midst of customers while delivering high-quality real time multimedia content. While accessing the service by a customer, it is possible to adapt real time streams to vastly changeable network conditions to give suitable quality in spite of factors upsetting Quality of service. To reach these intentions, adaptive web service processes to supply more information for determining the quality and size of the delivered object.

Keywords: *Enterprise Network, Quality of Service, Content Delivery, Real time streaming Protocol.*

I. INTRODUCTION

Applications of CDNs can also be found in many communities, such as academic institutions, advertising media and Internet advertisement companies, data centers, Internet Service Providers (ISPs), online music retailers, mobile operators, consumer electronics manufacturers, and other carrier companies. Along with the proliferation, formation, and consolidation of the CDN landscape, new forms of Internet content and services are coming into picture while distribution and management of content is introducing new challenges in this domain. This raises new issues in the architecture, design and implementation of CDNs. The technological trends in this domain need to be explored in order to provide an exclusive research roadmap to the CDN community. The Real Time Streaming Protocol (RTSP) is a network control protocol designed for use in entertainment and communications systems to control streaming media servers.

The protocol is used to establish and control media sessions between end points. Clients of media servers issue VCR-like commands, such as *play* and *pause*, to facilitate real-time control of playback of media files from the server. The transmission of streaming data itself is not a task of the RTSP protocol. Most RTSP servers use the Real-time Transport Protocol (RTP) for media stream delivery; however some vendors implement proprietary transport protocols. The RTSP server from Real Networks, for example, also features Real Networks' proprietary RDT stream transport. Web services can also be used to implement architecture according to Service-oriented architecture (SOA) concepts, where the basic unit of communication is a message, rather than an operation. This is often referred to as "message-oriented" services.SOA Web services are supported by most major software vendors and industry analysts. Unlike RPC Web services, loose coupling is more likely, because the focus is on the "contract" that WSDL provides, rather than the underlying implementation details. Middleware Analysts use Enterprise Service Buses which combine message-oriented processing and Web Services to create an Event-driven SOA. [6]At the dawn of the third millennium a new breed of web application has risen: Web Services (WSs) These services are "self-contained, self-describing, modular applications that can be published, located, and invoked across the Web.Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service." Since they first appeared, several research groups have worked on building efficient frameworks that enable the deployment of web services, exploiting technologies such as : XML (Extensible Markup Language), SOAP3 (Simple object Access Protocol), UDDI(Universal Discovery, Description and Integration),WSDL5 (Web Services Description Language), SOA (Service Oriented Architecture) , etc.

Some of these groups have focused on developing ontologies that capture the WSs' main properties. Nevertheless, little work has been done to represent the non-functional features of WSs, the most critical part of which concerns their Quality of Service (QoS). Integrating QoS features in the profile of WSs is to the advantage of both users and providers. QoS profiles of WSs are crucial in determining which service best addresses the user desires and objectives. If the discovered WSs are accompanied with descriptions of their non-functional properties, then the automated WS selection and composition that takes place, considers the user's QoS preferences in order to optimize the user's WS-experience regarding features such as performance, reliability, security, integrity, and cost. On the other hand, QoS can give WS providers a significant competitive advantage in the e-business domain, as QoS aware services meet user needs better and thus attract more customers. Adopting a WS best effort policy that does not provide any guarantees on response time, security, throughput, or availability, may still be acceptable in light, non-time-critical and non privacy-sensitive WSs (e.g., static weather forecast report service); it is, however, totally unacceptable in more demanding cases, when for example dynamic composition of various heterogeneous WSs is required. Moreover, QoS awareness in WS provision, coupled with dynamic network resource allocation mechanisms, enables providers to maximize the utilization of their infrastructure, thus contributing to the increase of their profits.

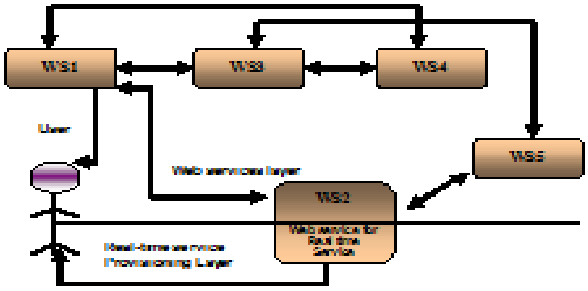
Lately, some research teams, having identified the importance of QoS featured WS profiles, have started to work on building QoS ontologies for web services, mainly focusing on developing ontology vocabularies, i.e., identifying the various QoS ontology parameters that are involved in web service provision However, as QoS parameters can be a lot more than type-value pairs, the need to develop a uniform way to efficiently represent the plethora of information concerning QoS parameters in a machine interpretable manner, while supporting enhanced reasoning functionalities, has appeared. A QoS ontology language that provides a standard model to formally describe arbitrary QoS parameters and exhibits properties such as completeness, flexibility, interoperability, reliability, scalability and accuracy. This language, combined with the proposed vocabulary, formulates a robust QoS semantic framework for WSs that can increase both the users' satisfaction and the providers' gains. The aim of this review focuses to the design of a framework to enable the QoS analysis of Web-service processes for real-time service provisioning (RTSP) based on service compositions. An integrated approach to quality of service for content delivery using Web services includes

a. Quality definitions for the framework model
b. User contracts
c. Fault monitoring System
d. Security measures and also
e. QoS broker design that can be used in providing QoS Web services efficient;
f. The end-to-end QoS issue for Web service composition;
g. The Several complex service selection algorithms to be used by QoS brokers.

II. QOS FRAMEWORK – A REVIEW

A. Web services and streaming delivery:

The framework deals with processes interacting with different actors and offering value added services that are able to satisfy user requests for complex objects, such as an e-learning object, a clinical health service, or an e-government service. The methods of quality analysis and the reference-tool architecture that combine the worlds of Web services and streaming by focusing on jointly provisioning complex services and their quality. We assume that the environment is composed of several nodes operating at two layers: Web services and their related protocols, and RTSP protocols. Figure 1 shows the reference scenario: a user requires, and eventually receives, a complex service obtained as a composition (possibly a choreography3) of different Web services; one of these (WS2 in the figure) provides streaming content. The main concerns of this review include addressing problems associated with the guarantee of QoS requirements in variable contexts and providing an active approach to solving or anticipating possible failures. Therefore, the focus of this article is not only on monitoring, but also on anticipating faults with predictiontechniques



.

Figure 1 Reference Scenario

B. QoS definition:

Defining a general QoS model is essential. Normally there are two QoS models, one for the Web service layer and one for the RTSP layer. The use two ontologies to represent quality parameters, with the semantics conforming to methodologies and techniques used in the Semantic Web community. The OWL Web Ontology Language [8] to develop the QoS ontologies, and followed the conceptual structure proposed by Papaioannou. Web-service QoS model relies on the following parameters to describe the QoS related to a single synchronous operation provided by the server:

a. ***Response Time***: Time elapsed between the instant a request is sent from the client and the instant the server computes the response.

b. ***Price***: Amount a client pays to the server for operation provisioning.

c. ***Availability***: Probability that a given operation is accessible at the moment of the request.

d. ***Reputation***: Ratio of the number of invocations with the requested QoS and the total number of invocations.

e. ***Data Quality Timeliness***: Freshness (up-to-date degree) of data.

f. ***Data quality accuracy***: Correspondence between given data and reference data considered as correct.

g. ***Data quality Completeness:*** Coverage of exchanged data with regard to total data representing the managed information.

C. Real-Time Service Provisioning Layer:

The QoS of a multimedia stream is based on two classes of parameters, namely:

a. User related - These express the user's requirements and preferences in accessing multimedia services, and allow the evaluation of relevance to the user of each component (video, audio, and data) of the delivered multimedia flow.
b. Network related - These parameters support the assessment of the amount of available network resources (bandwidth, channel speed, and so on).

D. User Contracts:

A QoS contract between a provider (server) and a consumer (user) regarding a set of parameters. User contract consists of two parts:

a. Mandatory part - consisting of the seven levels of QoS one for each QoS parameter and of a rule used to determine whether a QoS violation occurs;
b. Optional part - related to specific aspects, such as the QoS of real-time contents provided by the server.Table1 provides the entire set of parameters defining the QoS at the RTSP layer.

Table 1 RTSP Layer QoS Parameters

Parameter	QoS Parameter
User Related	Access count Video access count Audio access count Data access count Video degradation count Audio degradation count Data degradation count *x*-resolution *y*-resolution Chrominance Luminance Frame rate Audio channels Audio codex Audio frequency
Network Related	Video bandwidth Audio bandwidth Data bandwidth

E. QoS for web services:

Future Web-based systems require a seamless integration of user processes, server applications, domain intelligence, and Web services over the Internet. Delivering QoS services for most multimedia and real-time applications is a critical and significant challenge because of the dynamic and unpredictable nature of user applications and Internet traffic.

User applications with different profiles and requirements compete for the resources used to provide Web services. Without a careful management of Web service QoS, critical applications may suffer performance degradation, and resulting in customer dissatisfaction or media losses. The area of QoS management covers a wide range of techniques that match the needs of service requestors with those of the service provider's. QoS has been a major area of study in computer networking, real-time computing , and system middleware. For Web services, QoS guarantee and enhancement have started to receive some attention. The proposed work only consider the following quality attributes as part of Web service parameters.

a. Response time (T): The amount of time to get a service request fulfilled at the client side. This includes service time Ts and transmission time Tt: T = Ts + Tt .
b. Service time (Ts): The time a server needs to process a service request. The information is furnished by the service provider;
c. Transmission time (Tt): The time needed to send a request to a server and get the result from the server (i.e., round trip communication time). It is decided by the network.
d. Cost (C): Includes service cost Cs and transmission cost Ct: C = Cs + Ct.

   Service cost (Cs): Service cost is the service charge for each unit of service. A Web service may be priced differently depending on the quality of the (media) service delivered. It is set by the service provider;

   Transmission cost (Ct): The price that a service requestor has to pay for transmitting a request to a server and transmitting the result data from the server to the requester. The transmission cost is decided by the network operator;
e. Service availability (A): The probability that a service is available at some interval of time. This only measures the server availability in terms of responding to a request, not the result quality. It can be computed from historical data: A = Ta / Tt where Ta: the amount of time that a service is available;Tt: total time interval that is measured
f. Service reliability (R): The probability that a request is correctly fulfilled within the expected time. It can be computed from historical data: R = Ns / N where Ns: number of requests successfully fulfilled;N: total requests.
g. Network bandwidth: The minimum network bandwidth required to receive the service. This is especially important for services with multimedia content such as video or large graphics. The bandwidth attribute will also be important for Web service brokers to decide if a service should be invoked if the client is using a low bandwidth network such as wireless connections. The above QoS attributes that the work is consider in the framework are both easy to understand and to measure. These attributes can be collected on a system without user intervention. For example, before and after each connection and invocation of a Web service, a software agent can automatically measure the response time, the service cost, the bandwidth used, and the number of connection attempts before the service is successfully delivered.

F. Prediction:

To anticipate faults, the proposed work uses a prediction model and a support framework based on monitoring and machine learning. Because the global QoS varies in the runtime environment, to determine the global QoS by observing a set of parameters (the prediction global QoS is the tuple containing only the Web-service quality parameters).Some regularity can emerge from observation of the global QoS, depending on the values of specific parameters in different situations.

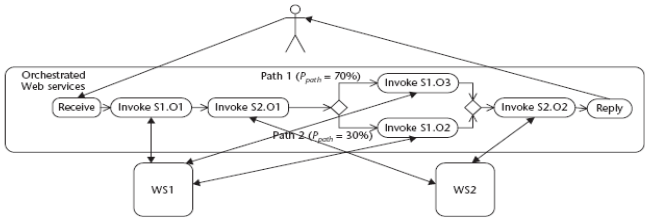


Figure 2. Sample Web services process for QoS prediction

For example, the global QoS on the same sequence of operations can change, and this regularity can be useful in determining the range variability. By observing these regularities, it can be define or predict the global QoS. In this scenario, a huge set of simulated data contained in the log files generated by the runtime environment is available, making it possible to analyze this data to define behavioral models. The model uses machine-learning techniques to build a system capable of providing suggestions on possible variations of the global QoS.Fig 2 shows a sample process useful in describing the learning problem. A process instance uses two services S1 and S2, and invokes several operations (O1, O2, . . ., On). To formalize the learning problem as follows: given an answer to a process operation Si.Oi and given the current global QoS level $Curr_{QoSLSi:Oi}$, let try to know with a certain probability $P_{globalQoS}$, the global QoS level global-QoS_j corresponding to $Curr_{QoSSSi:Oi}$, with j being an identifier of a future operation of S_i. The following features represent input instances for the classifier:

a. Service name S identifying the server providing the current operation;
b. Operation name O identifying the current operation;
c. QoS parameters T.Resp, P,Avail, Rep, DQ. Timel, DQ. Acc, and, DQ. Compl identifying the current values of each of the seven QoS parameters; and
d. Target operation O_{Target} identifying the next operation upon which the QoS prediction will be performed.

G. QoS framework:

The architecture monitors, detects, and predicts QoS faults, which the architectural components detect and manage by providing a self healing Web-service approach. The approach consists of making the services aware of possible faults and capable of repairing them. The business scenario consists of several coordinated services that provide real-time content. Process management occurs on the communication layers for Web-services interaction and for RTSP due to the technological differences between the two layers. However, it's necessary to manage the two layers uniformly from the user perspective, and hence the streaming server exposes a management interface to the Web-service layer. The separation is total for communication protocols, while for QoS, information exchange is enabled between the two layers to react to QoS faults.

H. Web-service Layer

In the Web-service layer, monitoring involves several purposes:

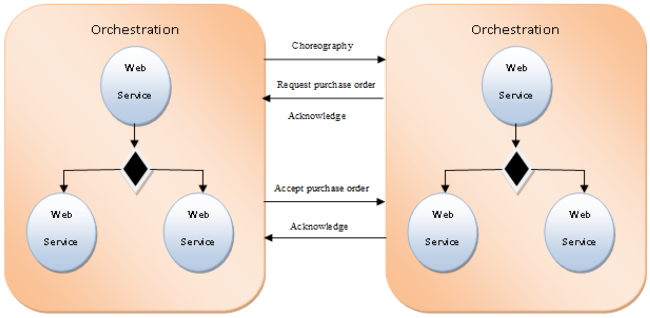


Figure 3 Orchestration versus choreography

a. Checking if the execution of the complex service correctly follows the interaction protocol defined by the global choreography
b. Checking whether a QoS contract is respected;
c. Estimating the QoS of following operations to prevent QoS faults.

Fig 3 shows Web-service choreography, none of the involved participants centrally executes the composite service. Each Web service is simply aware of its own status and doesn't have a global view encompassing all the cooperating services. However, the choreography definition represents global perspectives on the composite service that a choreography-monitoring Web service can rely on to detect faults occurring at the choreography level. The choreography monitor, relying on the notification messages received by each Web service and on the global choreography description, can track the progress of the service execution. Thus, the choreography monitor can detect possible mismatches between the order of message exchanges occurring during service execution and the one prescribed by the choreography definition.

I. Real-time service provisioning layer:

Monitoring bandwidth available for RTSP plays a key role in the processes related to multimedia streaming. Specifically, knowledge of the bandwidth available on each network link enables the detection of bottleneck links. Monitoring bandwidth is therefore essential for regulating and improving the QoS associated with a streaming application. Table 2 refers the concept of global QoS as a tuple composed of the union of different QoS values or levels, due to the heterogeneity of the range domains.

III. EMPIRICAL COMPARISONS

The following table 2 shows the performance of the two frameworks

Table 2 Empirical Comparisons – QoS Framework for CDN using Web Services

	Web Services and Streaming delivery framework			**QoS Ontology Framework Ambient QoS**		
	Input	Parameter	Values	Input	Parameter	Values
QoS Framework for CDN using Web Services	Current QoS	Response time Price Availability Reputation Data quality timeliness Data quality accuracy Data quality completeness	5 s 7 Euros 0.9 0.75 0.75 0.75 0.75	Current QoS Contract	Response Time Price Availability Capacity Scalability (max) Jitter (max) Error Rate (max) Latency (min) Throughput (Kbps)	6s 20 Euros 0.88 200 0.80 1 (msec) 10-5 300(msec) 384

The following graph shows the performance of Ambient QoS Framework.

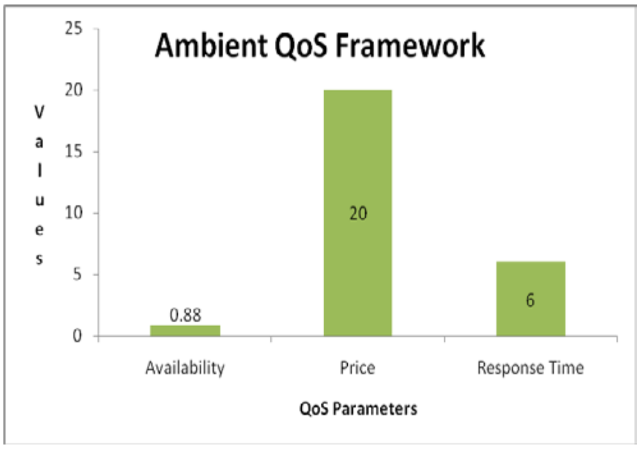


Figure 4. The performance graph – QoS Ontoloy framework

The following graph shows the performance of QoS Framework with streaming delivery

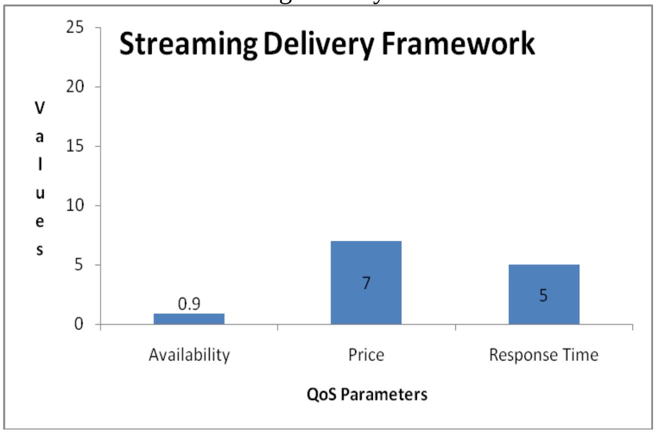


Figure 5 The performance graph – QoS framework with streaming delivery

IV. CONCLUSION

The proposed model has End-to-End management infrastructure for applications, systems, and network. It gives them flexible control over business processes involving Web services. Content Delivery Networks (CDNs) address the problem of network congestion by storing and serving internet content from different distributed locations rather than from a few centralized origin points. The goal of QoS framework is to provide guarantees on the ability of a network to deliver predictable results. The network QoS refers to the ability of the network to handle the traffic such that it meets the service needs of certain applications.

V. REFERENCES

[1] Francesco Buccafurri, Pasquale De Meo, Gianluca Lax, and Domenico Ursino university Mediterranea of Reggio Calabria, Mariagrazia Fugini, Stefano Modafferi, and Barbara Pernici Politecnico di Milano, "A Framework for Using Web Services to Enhance QoS for Content Delivery" IEEE Transactions on Intelligent and pervasive Multimedia systems, 2010.

[2] M. Pathan, R. Buyya, and A. Vakali, eds., Content delivery Networks, LNEE 9, Springer, 2008.

[3] F. Buccafurri et al., ''Analysis of QoS in Cooperative Services for Real Time Applications,'' Data and Knowledge Engineering, vol. 67, no. 3, 2008, pp. 463 484.

[4] D. Ardagna et al., ''PAWS: A Framework for Processes with Adaptive Web Services,'' IEEE Software, vol. 24, no. 6, 2007, pp. 39 46.

[5] J. Sokol and K.P. Eckert, ''MCDN: Multimedia Content Discovery and Delivery,'' Proc. Int'l Symp. Autonomous Decentralized Systems, (ISADS), IEEE Press, 2007, pp. 411-420.

[6] L. Ardissono et al., ''Monitoring Choreographed Services,'' Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, T. Sobth, ed., 2007, Springer Verlag, pp. 283 288.

[7] T. Yu and K.J. Lin, ''QCWS: An Implementation of QoS-Capable Multimedia Web services,'' Multimedia Tools and Applications, vol. 30, no. 2, 2006, pp. 165-187.

[8] I.V. Papaioannou et al., ''A QoS Ontology Language for Web-Services,'' Proc. IEEE Int'l Conf. Advanced Information Networking and Applications,(AINA), IEEE CS Press, 2006, pp. 101-106.

[9] I.H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed., Morgan Kaufmann Publishers, 2005.

[10] M. Little, E. Newcomer, and G. Pavlik, eds., Web Services Context Specification (WS-Context), OASIS committee draft, Nov. 2004; http://xml.coverpages. Org/WS-ContextCD-9904.pdf.

[11] D.L. McGuinness and F. van Harmelen, eds., OWL Web Ontology Language Overview, World Wide Web Consortium (W3C) recommendation, Feb. 2004;

[12] WS-Diamond team, ''WS-Diamond:WebServicesDIAgnosability,MONitoring, and Diagnosis,'' to be published in At Your Service: An Overview of Results of Projects in the Field of Service Engineering of the IST Program, Series on Information Systems, MIT Press.

[13] W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, ''Business Process Management: A Survey,'' Proc. Int'l Conf. Business Process Management, (BPM), LNCS 2678, Springer, 2003, pp. 1-12.

[14] M. Jain and C. Dovrolis, ''End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput,'' IEEE/ACM Trans. Networking, vol. 11, no. 4, 2003, pp. 537 549.

[15] G. Araniti et al., ''Adaptively Controlling the QoS of Multimedia Wireless Applications through User Profiling Techniques,'' J. Selected Areas in Comm., vol. 21, no. 10, 2003, pp. 1546-1556.

Short Biodata of the Author

K.Manivannan received B.E degree in Computer Science and Engineering from the Anna University, Chennai and M.E degree in Computer Science and Engineering from the same University. He is currently working toward the Ph.D degree in Computer Science and

Engineering at the Anna University of Technology, Madurai.His current research interests include web services, distributed systems and network architecture.

C.G.Ravichandran received B.E degree in Electronics and Communication Engineering from Bharathiyar University in 1988 and M.E degree in Electronics Engineering from Anna University, Chennai in 1991.He received the Ph.D degree in Medical Image Processing from Anna University, Chennai in 2009.Currently he is a Principal of RVS College of Engineering and Technology Dindigul,Tamilnadu,India. His research interests include image segmentation, medical image processing, network architecture, distributed systems and web services.