Quality of Service in the OSA/Parlay Environment

Y. Bata and H.E. Hanrahan
Centre for Telecommunications Access and Services (CeTAS)[1]
University of the Witwatersrand
Johannesburg, South Africa
e-mail: {y.bata, h.hanrahan}@ee.wits.ac.za

Abstract— Application related Quality of Service in an open service environment requires a reliable Quality of Service framework. Apart from the Connection Management Service Control Function, OSA/Parlay lacks data structures and mechanisms for supporting the entire Quality of Service (QoS) cycle. This paper establishes the need to develop a QoS framework for OSA/Parlay in which QoS requested by an end-user is fulfilled by the network.

Index Terms—OSA/Parlay, QoS Framework

I. INTRODUCTION

NETWORK Operators and Service Providers are looking for new sources of revenue, and new value added services are a key part of this initiative. The OSA/Parlay enables the development of these new services using the same technologies used for rapid application development in the IT community: open APIs, distributed computing, Java and Web Services [1]. Parlay essentially provides interfaces (called OSA/Parlay APIs) for applications with which they can use to access and control certain aspects of telecommunications networks. The OSA/Parlay APIs are network independent: they are designed to be used for mobile networks, for fixed networks and for next generation networks based on the IP protocol [1]. The OSA/Parlay APIs also provides software technology independence by providing a generic, technology-independent interface.

The OSA/Parlay architecture is based on a three-layered architecture shown in figure 1 consisting of:

- The Services & Applications Layer: consists of the application logic (located in an Application Service Provider domain [2])
- The Service Control Layer: contains generic, stable and robust functionality to support real-time and information services using the Transport Network Layer
- The Transport Network Layer: provides services to support bearer connections as requested by the Service Control Layer

[1] The Centre is supported by Telkom SA Limited, Siemens Telecommunication and the THRIP Programme of the Department of Trade and Industry

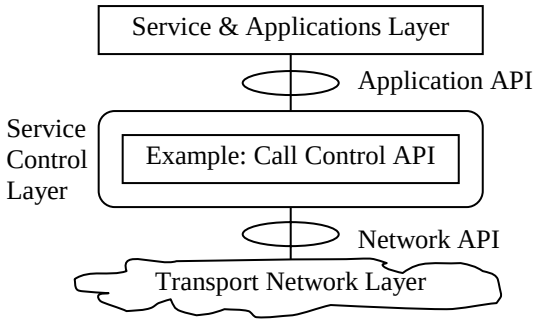

Figure 1: OSA/Parlay Architecture

Each layer is independent of its neighbouring layer and is accessed through an open, standard and secure API [3]. This architecture facilitates the integration of various underlying networks and provides a single interface (the OSA/Parlay API) to provide services on these networks [3].

Using the API approach in communications means that it is possible to write portable applications that will run on a variety of underlying protocols without change. For example, a developer can build an OSA/Parlay application that sets up a conference call when a group of people are free, while not restricting the application to an IS-41 network, a GSM network or a SIP-based network [1]. This example application would use one of OSA/Parlay's comprehensive set of APIs built for communications applications, namely the Call Control API. The Call Control API allows applications to set up calls in the network, set up multi-party calls in the network, route calls from the network and set up multi-media calls. Upon request for a service from an application, an API (see figure 1) uses the Network API (see figure 1) to fulfil the request in the underlying network (without concern for the actual type of underlying network). The Network API and its required functionality are not defined by the OSA/Parlay standards, but by implementations of the standard.

As OSA/Parlay develops towards a better service provision infrastructure, certain areas critical to future operation and general acceptance of OSA/Parlay as the future service delivery platform still remain unresolved [4]. One of the areas that need to be addressed is the provision of support for Quality of Service (QoS). Apart from the Connection Management Service Control Function (in the Connectivity Manager API), OSA/Parlay lacks data structures and mechanisms for supporting the entire QoS cycle through it's inform, negotiate, establish, operate and release phases [4]. This paper proposes the development of a QoS framework for OSA/Parlay that supports QoS on a per-user, per-application basis.

II. What does QoS mean?

QoS is described differently at different levels. For users of applications (at the User Level) a high-level description of QoS is used. This is normally in the form of a set of user-friendly categories, like the Olympic Service Profile [5] where a user can choose a Gold, Silver or Bronze QoS. At the Network Level however, low-level descriptions of QoS are used (normally in the form of parameters). Examples of low-level parameters describing QoS at the Network Level are bit rate, delay and jitter. These are very technical descriptions and are seldom of any concern to users at the User Level. Various other descriptions of QoS are used at different levels in the architecture [6]. In developing a QoS framework for OSA/Parlay, the description and understanding of QoS at various levels in the OSA/Parlay architecture must be taken into account.

III. Quality of Service Framework

A number of standardisation bodies are currently dealing with the complex area of QoS for communication systems in general. Communication recommendations are however specific to a particular technology, and do not consider QoS implications for architectures combining a number of communication technologies [6]. These considerations are very important for the OSA/Parlay framework since it targets to be a framework for integrating any service and transport technology in a service and technology independent manner. Consequently, it has to deal with QoS for all kinds of services communicating via a wide range of technologies [6]. A well-defined QoS framework is therefore crucial for the success of OSA/Parlay.

IV. Requirements

The objective of the proposed research is to develop a QoS framework for OSA/Parlay in which QoS requested by an application is fulfilled by the network. Development of the framework includes the development of a data structure that supports the different descriptions of QoS at different levels in the OSA/Parlay architecture. It must also provide mechanisms to translate and transfer these descriptions across different levels in the architecture in order to support QoS effectively. A QoS framework that supports all kinds of services communicating via a wide range of Network Layer technologies must be considered.

V. Possible Approaches

The TINA architecture proposes the use of a layered QoS framework, with QoS mapping between layers. This may be an effective approach to developing QoS framework and must be further investigated. Another possible approach is to define a completely new layer – the Resource Control and Management Layer – under the Service Control Layer, which could provide a further level of abstraction from the Network Layer [2], and therefore make the provisioning of QoS to applications a simpler task.

VI. Conclusion

Apart from the Connection Management Service Control Function (in the Connectivity Manager API), OSA/Parlay lacks data structures and mechanisms for supporting the entire QoS cycle through it's inform, negotiate, establish, operate and release phases. In order for OSA/Parlay to be a viable future architecture, the issue of QoS must be urgently addressed. This paper proposes the development of a QoS framework for OSA/Parlay that supports QoS on a per-user, per-application basis. A possible approach to developing the required framework includes adapting the TINA architecture's layered QoS framework concept. Another approach that must be considered is the introduction of a new layer that makes the provision of QoS to applications a simpler task.

VII. References

[1] Lozinski Z., "Parlay/OSA – a New Way to Create Wireless Services", http://www.parlay.org/docs/2003_06_01_Parlay_for_IEC_Wireless.pdf, last accessed 2nd June 2005.

[2] Hanranhan H.E., "*Convergence: Networks, Services and Applications*", CeTAS, 2005.

[3] Sperryn W.J., "*Performance monitoring in the TINA Distributed Processing Environment*", University of the Witwatersrand, Johannesburg, December 2000.

[4] Mwansa D., Hanrahan H., "*Conceptual QoS Framework for Managing End-User Services in the Next Generation Network*", CeTAS, 2004.

[5] CeTAS, "*Research Project Plan for 2005*", 19th October 2005.

[6] Rajahalme J. et al, "*Quality of Service Negotiation in TINA*", IEEE, 1998.

Yusuf Bata is presently enrolled in his first year of an MSc (Electrical Engineering) degree in Telecommunications at the University of the Witwatersrand. He completed a bachelor's degree in Electrical Engineering (Information Engineering) at the University of the Witwatersrand in 2004.