Position paper in *Proceedings of the Sixth International Working Conference on Dependable Computing for Critical Applications (DCCA-6)*, IFIP, Grainau, Germany, March 1997. Published in the conference book *Dependable Computing for Critical Applications* 6, editors M. Dal Cin, C. Meadows, W. Sanders, IEEE Computer Society, 1998, p. 149– 152. This was background material for a panel Bakken chaired at this conference.

On Specification, Metadata, and Binding of Multi-Property Quality of Service[1]

David E. Bakken
BBN Systems and Technologies, Cambridge, Mass., USA
dbakken@bbn.com

Abstract

This position paper offers three issues which must be addressed for successful multi-property quality of service (QoS) middleware. First, specifications must be at a high level, qualitative, and with flexible ranges. Second, an infrastructure for QoS metadata is crucial but almost totally lacking. Third, late binding is crucial.

1. Introduction

In the last few years the notion of quality of service (QoS) has been discussed with increasing frequency. The preponderance of QoS research has focused on network communications parameters for multimedia applications specified at connection setup, at the socket level, with absolute "take it or leave it" guarantees, and with no or few provisions for adaptivity [1,4,7]. This is understandable, because many believe that there is a lot of money to be made in the next few years delivering "video on demand." However, it is unfortunate, because the QoS concept is much broader than this limited scope [2].

The last few years have also seen a remarkable increase in the number of distributed applications and infrastructure to run them on, including the World Wide Web, CORBA, DCOM, and similar technologies. Indeed, by the time of this publication perhaps a million users will have a CORBA ORB on their desktop as part of their Netscape 4 browser. Further, many distributed applications used by the military for command and control, logistics for peacekeeping and wartime, and other purposes operate in hostile environmental conditions similar to or even worse than today's overloaded Internet.

The purpose of this panel, "Extending QoS to include Performance, Dependability, and Security," is to explore the issues surrounding the question "Can QoS paradigms and architectures be used to support these more general

[1] This research was funded in part by the Defense Advanced Research Projects Agency (DARPA) under Rome Laboratory Contract No. F30602-96-C-0315.

applications in diverse environments?" That is, can QoS be used to integrate architectural support for other properties, environmental conditions, and application domains with more adaptivity than today's systems and architectures offer? If so, what must be done to see this vision happen? Clearly, this vision is as important to achieve as the technical hurdles are difficult [3]. This position paper offers issues which we feel must be addressed for this vision to occur.

2. Observations on achieving multi-property QoS

In the last few years we have been developing an object-oriented QoS architecture intended for wide-area networks (WANs) supporting multiple properties, with adaptivity, but without absolute guarantees [9]. From this experience I offer some observations about QoS specifications, QoS metadata, and late binding, respectively.

2.1 QoS specifications

Distributed application developers have a hard enough time building and maintaining a distributed application over its lifetime without considering QoS issues. Indeed, many of them have little idea of their invocation patterns let alone their QoS requirements. It is thus beneficial if the QoS contract elements can be at a high level of abstraction and qualitative with a flexible range of parameters.

These high-level and fuzzy specifications must then be translated one or more times to reach an actual implementation (e.g., a Horus protocol stack [6]). For example, if one wanted to offer contract abstractions of network performance and availability reservations, one could use paradigms such as active replication or the primary/backup approach, choosing the one which best matches the client's requirements. The QoS middleware developer then has multiple implementations of each paradigm, and has modeled and evaluated each thoroughly so that he knows their performance and resource usage. The implementation best suited for current conditions is then used, which would include some form of replica management and bandwidth reservation based on the client's promised usage patterns, required availability, and the object's internal communications (for replication management).

2.2 QoS metadata infrastructure

Any QoS framework needs to have *QoS metadata*, data about runtime conditions such as current capacity, failure status, traffic, response requirements, and budget requirements. These data are required at many levels in a system to choose the best implementation, route, etc., given the current conditions and client requirements. It is crucial to get such metadata in an organized and simplified way to help overloaded programmers and to facilitate reuse; the more QoS properties which are supported the more urgent this becomes.

These metadata are provided by a number of means. Some are sent via *in-band* means, in the same network packet as the functional invocation is. An example of this is the Inter-ORB Reference (IOR) of the CORBA Internet Inter-ORB Protocol (IIOP) [5]. IORs are presently only used for authentication purposes, but we consider them an excellent mechanism for passing in-band QoS metadata and hope to use them for such in the near future. Other QoS metadata are passed on

the same path but *side-band*, in a separate packet from the functional invocation. RSVP uses this scheme [8]. Finally, some information is passed *out-of-band*, across an entirely separate path from any functional invocations a given client is making. Examples of this include capacity information from a network management system, a QoS MIB, and a failure detector.

It is unrealistic to believe that every application writer or even QoS substrate developer can develop such an infrastructure for their one use to integrate the metadata, given that they come from many sources, at many times, and from many places, as discussed above.. There is thus a great need to have an infrastructure to collect, organize, disseminate, and simplify these QoS metadata The current state of such infrastructure is greatly lacking, and this may be one of the biggest impediments to realizing multi-property QoS, or even any QoS across a WAN.

2.3 Late Binding

It is crucial for QoS middleware to support a spectrum of binding times, including as late binding as possible. Binding decisions are made assuming a set of environmental and other conditions, and if they are made prematurely then those assumptions can be wrong, resulting in poor performance. The later the decision the more accurate the information and the more flexible the system. Further, if a program cannot easily reconfigure (via late binding) in the face of a security breach or other attack, then its survivability is limited. Finally, if decisions are bound at design time then it can be much harder for the program to evolve to be used in different (sometimes unforeseen) environments, a crucial need for today's military applications. Late binding is thus an important architectural feature which can be quite useful when employed judiciously.

References

[1] Aurrecoecha, C., Campbell, A. and Hauw, L. A survey of QoS architectures. *Multimedia Systems Journal,* Special Issue on QoS Architectures, 1996.

[2] ISO, Quality of service framework. ISO/IEC JTC1/SC21/WG1 N9680, International Standards Organization, UK, 1995.

[3] McLean, J. and Heitmeyer, C. *High assurance computer systems: a research agenda*. America in the Age of Information, National Science and Technology Council Committee on Information and Communications Forum, Bethesda, Maryland, USA 1995. See also `http://www.itd.nrl.navy.mil/ITD/5540/HighAssurWkshp/reports.html`

[4] Nahrstedt, K. and Smith, J. The QoS Broker. *IEEE Multimedia*, 2(1), Spring 1995, 53-67.

[5] Object Management Group,. *The common object request broker: architecture and specification.* OMG Document 96-03-04, July 1995.

[6] van Renesse, R., Birman, K. and Maffeis, S., Horus: A flexible group communications system. *Communications of the ACM*, 39(4), April 1996.

[7] Vogel, A., Kerherve, B., Bochmann, G., and Gecsei, J. Distributed multimedia aplications and quality of service: a survey. In *Proceedings of the First IEEE Metadata Conference*, Silver Spring, Maryland, USA, April, 1996.

[8] Zhang L., Deering, S., Estrin, D., Shenker, S., and Zappala, D. RSVP: a New Resource ReSerVation Protocol. *IEEE Network*, 7(6), September 1993, 8-18.

[9] Zinky, J., Bakken, D. and Schantz, R. Architectural support for quality of service for CORBA objects. *Theory and Practice of Object Systems*, Special Issue on the OMG and CORBA, 3(1), April, 1997, to appear. See also `http://www.dist-systems.bbn.com/papers/97/TAPOS/QuOTAPOS.ps`.