

**Techniques for Implementing
Quality of Service
In the Eclipse, Nemesis and Real-Time Mach
Operating Systems**

Francesca Ruffolo

December 6, 2001

Instructor: Dr. Marius Soneru

CS 550
Comparative Operating Systems

Table of Contents

Abstract.....	3
1.0 Introduction.....	3
1.1 Eclipse Operating System.....	3
1.2 Nemesis Operating System.....	3
1.3 Real-Time Mach Operating System.....	4
2.0 Implementation of QoS in Operating Systems.....	4
2.1 Eclipse.....	4
2.1.1 Reservations.....	5
2.1.2 Tagging.....	5
2.1.3 Reservation Domains.....	5
2.1.4 Quality of Service Parameters.....	6
2.1.5 Move-to-Rear List Scheduling (MTR-LS).....	7
2.2 Nemesis.....	7
2.2.1 Quality of Service Manager.....	7
2.2.2 Run Time Resource Allocation.....	8
2.2.3 Accounting.....	8
2.2.4 Policing.....	9
2.3 Real-Time Mach.....	9
2.3.1 Processor Reserves.....	9
2.3.2 Dynamic QoS Server.....	9
2.3.3 Q-RAM (QoS-based Resource Allocation Model).....	10
3.0 Conclusions and Summary.....	12
4.0 Other Implementations of Quality of Service.....	12
5.0 References.....	13

Abstract:

The requirement to make quality of service (QoS) guarantees available is increasing with the need to run multiple server applications, such as audio and video media servers, web servers, and host services for multiple entities (organizations, individuals) on the same platform. This paper evaluates and compares the techniques used to implement quality of service in three separate operating systems: Eclipse/BSD, Nemesis, and Real-Time Mach.

1.0 Introduction

In order to support multiple real-time applications on a single platform, the operating system must have the ability to reserve system resources among several applications in order to achieve a certain level of performance. Although networks now provide quality of service guarantees with respect to packet delay and bandwidth, these are of little value if they are not extended directly to the application via the operating system.

The motivation for implementing quality of service (QoS) control in operating systems is to allow explicit control over provisioning of system resources among applications to achieve desired levels of predictable performance. In addition, quality of service is used to guarantee a certain level of performance to server applications.

The quality of service initiatives requires the changing of current operating systems in the area resource schedulers: CPU, disk (file systems), physical memory, and I/O.

1.1 Eclipse Operating System

Eclipse is a product of research that evolved from the Plan 9 operating system that was developed by Bell Laboratories in New Jersey starting in the late 1980's. Plan 9 was developed in response to several trends in the computing industry at that time.

- Networks of smaller, more personal machines were preferred over large centralized time-shared computers (typically UNIX workstations)
- Administrators grew weary of overloaded, routine time-sharing systems and wished to move to small, self-maintained systems

In the rush to accommodate customer's needs via the creation of personal workstations, weaknesses in doing so were overlooked. Plan 9 attempted to fix both problems.

- Graphics and networking that were added to UNIX in the latter part of its lifecycle were poorly integrated
- The early focus on private machines made it difficult for networks to serve as seamlessly as standard time-sharing systems.
- The current version of Eclipse, often referred to as Eclipse/BSD (based on FreeBSD (Berkeley Software Distribution) version 3.4), provides quality of service support for applications. Eclipse is utilized for guaranteeing QoS to server applications and distinguishing among performance of different web sites hosted on the same platform.

1.2 Nemesis Operating System

The Nemesis operating system results from work completed for the Pegasus and Pegasus II projects. The following universities and corporation partnered together in order to develop Nemesis:

- University of Cambridge, U.K.
- University of Twente, Netherlands
- University of Glasgow, U.K.
- Swedish Institute of Computer Science, Sweden
- Citrix Systems, LTD, U.K.