IP Quality of Service Support for Soft Real-Time Applications

Karthik Channakeshava[♣] Kaustubh S. Phanse[†] Luiz A. DaSilva[♣] Binoy Ravindran[♣]
Scott F. Midkiff[♣] E. Douglas Jensen[*]

[♣]Bradley Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, Virginia 24061 USA
{kchannak, dasilva, binoy, midkiff}@vt.edu

[†]Department of Computer Science and Electrical Engineering
Luleå University of Technology
SE-971 87 Luleå, Sweden
kphanse@sm.luth.se

[*]The MITRE Corporation
Bedford, MA 01730, USA
jensen@mitre.org

Abstract

To obtain acceptable timeliness performance for emerging large-scale distributed real-time control applications operating in large IP internetworks, a scalable quality of service (QoS) architecture is needed. In this paper, we propose a scalable QoS architecture (abbreviated as RTQoS) in support of real-time systems, one that implements real-time scheduling at end-hosts and stateless QoS in the core routers. We address challenges and explore potential benefits achieved by integrating network services with real-time systems, through the use of a network testbed. Experimental evaluation demonstrates the RTQoS architecture as a promising approach for soft real-time applications that are subject to time/utility function time constraints and utility accrual optimality criteria.

1. Introduction

Many large-scale distributed real-time applications are dynamic and asynchronous, and they often run over large multi-hop networks, including IP-based wide area networks or, in the future, wireless ad hoc networks. Given the performance-sensitive nature of these applications, quality of service (QoS) support from the network is crucial. There are a number of proposed solutions for providing network support for such applications over multi-hop networks. Most proposed solutions focus on achieving timeliness predictability and "hard" (timeliness) guarantees through the use of real-time channels, e.g., [1]. However, such solutions often lead to a stateful QoS architecture that lacks scalability.

The evaluation of a stateless approach to support QoS for real-time systems is still largely unexplored. In this paper, we propose a scalable IP QoS support model for soft real-time applications. We consider soft timeliness requirements for these applications that are expressed using time/utility functions (or TUFs) and utility accrual (or UA) optimality criteria. The proposed model, which we refer to as the *RTQoS architecture*, adopts a hybrid approach of using real-time packet scheduling in the network edge and policy-based stateless QoS support in the network core. The proposed model is schematically illustrated in Figure 1.

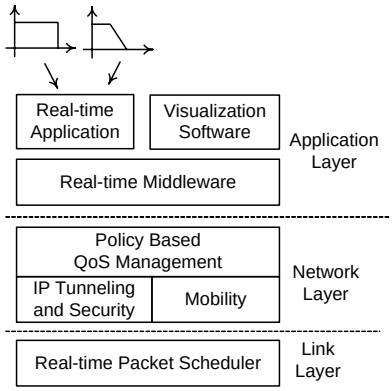


Figure 1: Proposed IP QoS and Network Model for Soft Real-time Applications

Our objective is, firstly, to assess the feasibility and benefits of integrating network services, such as policy-based management, network security, and quality of service mechanisms, with a real-time middleware; and, secondly, to evaluate how well the integrated system performs in providing QoS support for such soft real-time applications. We focus on the implementation and testing of our proposed model using an experimental network testbed. In our experiments, we consider the following three architectures:

(a) **NoRTQoS architecture:** No real-time scheduler and no QoS support in the network, i.e., best effort service;
(b) Our proposed **RTQoS architecture:** Real-time scheduler at end-hosts and stateless policy-based QoS in the core routers; and
(c) **RTRT architecture:** Stateful QoS with real-time scheduler on end-hosts and core routers.

We use the NoRTQoS and RTRT architectures as benchmarks, essentially two extremes on the QoS continuum, to compare the performance of our proposed architecture. Our results indicate that the RTQoS model provides acceptable real-time application performance and alleviates the scalability and feasibility problems of the RTRT model. To our knowledge, Choi, *et al.* in [2] have taken a similar approach for hard real-time systems and proposed *off-line* delay estimation and verification, and, *online* admission control and packet forwarding mechanisms. In contrast to the approach of Choi, *et al.*,

we propose a policy-based dynamic resource allocation scheme in the network core for supporting soft real-time applications characterized by TUF/UA constraints.

2. Time/Utility Functions and Utility Accrual Optimality Criteria

In this paper, we focus on real-time systems that operate in environments with dynamically uncertain properties. These uncertainties include transient and sustained resource overloads due to context-dependent activity execution times and arbitrary activity arrival patterns. When resource overloads occur, meeting deadlines of all application activities is impossible as the demand exceeds the supply. The urgency of an activity is typically orthogonal to the relative importance of the activity—e.g., the most urgent activity can be the least important, and vice versa. Hence when overloads occur, completing the most important activities irrespective of activity urgency is often desirable.

Deadlines by themselves cannot express both urgency and importance. Thus, we consider the abstraction of TUFs [3] that express the utility of completing an application activity as a function of that activity's completion time. We specify deadline as a binary-valued, downward "step" shaped TUF; Table 1 shows examples. Note that a TUF decouples importance and urgency—i.e., urgency is measured as a deadline on the X-axis, and importance is denoted by utility on the Y-axis.

Many real-time systems also have activities that are subject to *non-deadline* time constraints, such as those where the utility attained for activity completion *varies* (e.g., decreases, increases) with completion time (see [4][5] for examples). This is in contrast to deadlines, where a positive utility is accrued for completing the activity anytime before the deadline, after which zero, or infinitively negative utility is accrued.

When activity time constraints are specified using TUFs, which subsume deadlines, the scheduling criteria is based on accrued utility, such as maximizing sum of the activities' attained utilities. We call such criteria, UA criteria, and algorithms that optimize them, as UA algorithms. In this paper, we focus on non-increasing, unimodal TUFs as shown in Table 1—i.e., TUFs with a single optimal completion time, where the utility never increases as time advances.

3. Implementation and Testbed

In this section, we describe the implementation and experiments that were conducted for evaluating our proposed scheme. We have implemented a real-time client and server for generating the real-time traffic. The applications embed the TUF information in the IP Options field of the IP header and send the packet to the server across the network. A real-time middleware provides the application program interface (API) for using the real-time features of the underlying networking sub-system. In addition, we have implemented a utility accrual packet scheduling algorithm (UPA) [6] in the Linux kernel for packet scheduling. Scheduling decisions by the UPA are based on the heuristic that *maximizing the local utility might lead to an overall higher system-wide utility*. QoS provisioning was enabled on the intermediate routers for the RTQoS scenario using the Diffserv [7][8] implementation on Linux. The UPA scheduler is enabled on routers in the RTRT scenario. For the NoRTQoS scenario both schedulers are disabled.

3.1. Network Testbed

The testbed consists of two end hosts running real-time applications (characterized by different TUFs) and the real-time middleware. In addition, background traffic is generated to load the links. The volume of background traffic is kept the same for all experiments. A third host acts as the destination for the application traffic and measures and visualizes performance in terms of deadline misses, utility, and throughput. The end hosts are connected via routers. For the RTQoS scheme, the application sources mark the application packets with the appropriate Diffserv Code Point (DSCP) value. This marking is used for differentiating the application traffic in the intermediate routers based on the policies installed.

At the source, the real-time application registers its characteristics with the middleware that embeds them in the IP Options field and uses the information for scheduling at the source host. In the RTRT architecture, this information is used by the intermediate routers to perform packet scheduling. In all cases, it is used by the destination to measure performance.

4. Experimental Evaluation

The experiments are broadly divided into two categories based on application time constraints as:
(1) Homogeneous TUFs (all real-time applications are characterized by similar TUFs); and (2) Heterogeneous TUFs (real-time applications are characterized by dissimilar TUFs).

For each of these categories, experiments were conducted for the three scenarios, NoRTQoS, RT-QoS, and RTRT, described in Section 1. Table 1 shows the characteristics for the different real-time applications used in our experiments. In the TUFs shown, the symbols 'U' and 'D' represent the maximum utility value and application deadline, respectively. The application "App3" has the highest maximum utility value, followed by "App2" and "App1." We assign "weights" to the applications such that, in case of homogenous TUFs the weight is proportional to the maximum utility and, for heterogeneous TUFs weight is proportional to the area under the utility function curve, which allows us to

consider the "shape" of the TUF. These weights are used to determine the minimum bandwidth guarantees for each application based on the corresponding *TUF pseudo-slope* – defined as the ratio of the maximum utility to the deadline, and used for making scheduling decisions. In addition to the real-time applications, background traffic was added to ensure a fully loaded network when the total volume of real-time traffic generated was less than the available link bandwidth. The experiment was run multiple times for all three schemes, NoRTQoS, RTQoS, and RTRT, to obtain 95% confidence intervals.

Figure 2 shows the system-wide percentage of utilization accrued (%UA) and percentage of deadlines missed (%DM) for heterogeneous TUFs. The X-axis in each plot represents the rate of application generated traffic. The performance of the RTQoS and RTRT schemes is much better than that for the NoRTQoS scheme. For lower data rates, RTQoS performs slightly better than RTRT. From Table 1, we see that maximum bandwidth is allocated for the step TUF followed by the soft-step TUF and the linear TUF. Thus, for lower data rates, the step and soft-step TUFs contribute more towards the system-wide utility. In the case of RTRT, since there is no specific bandwidth allocation, all applications deteriorate, slightly reducing the system-wide utility obtained. This situation is reversed for higher data rates and RTRT performs much better than RTQoS.

Figure 3(a) and Figure 3(b) show the %UA obtained under the different schemes for step and soft-step TUFs, respectively. From Figure 3(a), we see that the RTQoS scheme results in a higher utility than the RTRT scheme for application data rates up to 72 kbps (until the link becomes overloaded). Once the application traffic generation rate exceeds the allocated bandwidth, the performance of the RTQoS scheme drops rapidly and eventually (beyond 72 kbps) the RTRT scheme results in higher utility. Since the application having a step TUF (App3) has the highest maximum utility value, its contribution to the system-wide utility is the largest. Thus, in Figure 3(b) we observe that RTQoS performs slightly better than RTRT between 40 kbps and 56 kbps. This can be attributed to the fact that RTRT, unlike RTQoS, does not explicitly reserve bandwidth. But, when the link is overloaded (beyond 56 kbps), RTRT's performance degrades gracefully, while RTQoS degrades faster. RTRT, RTQoS and NoRTQoS all perform the same for App1 (not shown) and the performance is worse when compared to App2 and App3.

The overall trend remains the same for homogenous TUFs. The performance degradation for the RTRT scenario is gradual in comparison to the other two scenarios. RTQoS performs better than NoRTQoS and exhibits performance that is similar to that for RTRT. Figure 4 shows the system-wide performance for homogeneous TUFs.

5. Concluding Remarks

In this paper, we presented the RTQoS architecture for achieving scalable QoS in support of soft real-time applications in IP internetworks. Using our prototype implementation, we demonstrated ways to integrate network services such as policy-based management and IP QoS with real-time applications. The performance of the real-time applications, measured in terms of the accrued utility and percentage of missed deadlines, was found to be much better using our proposed RTQoS architecture as compared to the NoRTQoS or best-effort scenarios. Establishing end-to-end real-time channels in the RTRT scenario provided better overall guarantees as compared to the RTQoS scenario. The trade-off is that each of the intermediate routers in the RTRT scenario must be equipped with a UPA-style real-time packet-level scheduling algorithm. Backbone routers in wide area networks generally encounter heavy traffic loads and packet-level scheduling is not practical and scalable. Furthermore, such architectures are not suitable for mobile networking environments.

Our experimental results provide empirical evidence that the proposed RTQoS architecture, which is sensitive to the application utility functions rather than just the average data rates, is a promising approach for soft real-time applications. An important part of our future work is to investigate the ability of our proposed architecture to provide timeliness predictability, e.g., assured lower bounds on both individual and collective accrued utility that are probabilistically satisfied.

Acknowledgment

This work was supported in part by the Office of Naval Research through the Navy Collaborative Integrated Information Technology Initiative.

References

[1] D. Ferrari and D. Verma, "A Scheme for Establishment of Real-time Channels in Wide Area Networks," *IEEE JSAC*, vol. 8, no. 3, pp. 368-379, April 1990.

[2] B. Choi, D. Xuan, *et al.*, "Scalable QoS Guaranteed Communication Services for Real-time Applications," *Proc. 20th ICDCS*, pp. 180-187, 2000.

[3] E. D. Jensen, C. D. Kocke and H. Tokuda, "A Time-Drivern Scheduling Model for Real-Time Systems," IEEE RTSS, pp 112-122, Dec 1985.

[4] R. K. Clark, E. D. Jensen, *et al.*, "An Adaptive, Distributed Airborne Tracking System", *IEEE WPDRTS*, LNCS, Springer-Verlag, vol. 1586, pp 353-362, April 1999.

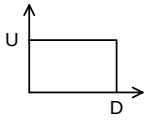
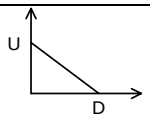
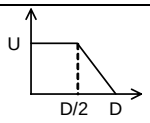
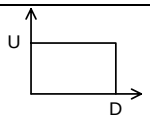
[5] D. P. Maynard, S. E. Shipman, *et al.*, "An Example Real-Time Command, Control, and Battle Management Application for Alpha," *Archons Project TR-88121*, Computer Science Department, Carnegie Mellon University, Dec. 1988.

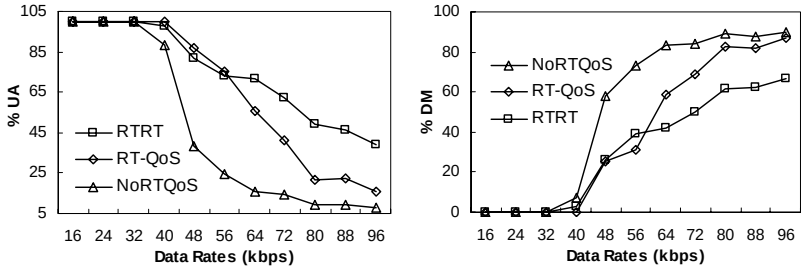
[6] J. Wang and B. Ravindran, "Time-Utility Function-Driven Switched Ethernet: Packet Scheduling Algorithm, Implementation, and Feasibility Analysis", *IEEE TPDS*, vol. 15, no. 2, pp 119-133, Feb. 2004.

[7] S. Blake, D. Black, M. Carlton, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *IETF RFC 2475*, Dec. 1998.

[8] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," *IETF RFC 2474*, Dec. 1998.

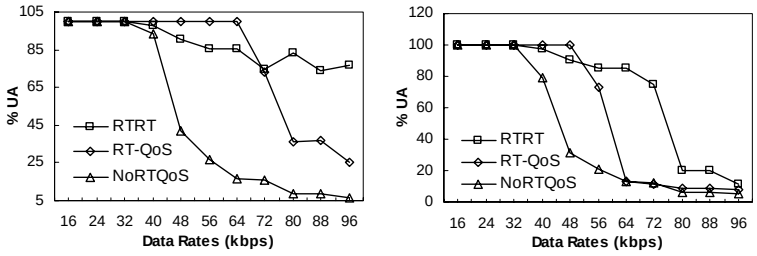
Table 1: Application Characteristics for the Experiments

Homogeneous TUFs				Heterogeneous TUFs			
TUF	Maximum Utility (U)	Weight	Minimum Bandwidth Guarantee (kbps)	TUF	Maximum Utility (U)	Weight	Minimum Bandwidth Guarantee (kbps)
	(App1) 100	1	22		(App1) 100	1	28
	(App2) 200	2	42		(App2) 200	1.5	43
	(App3) 300	3	64		(App3) 300	2	57



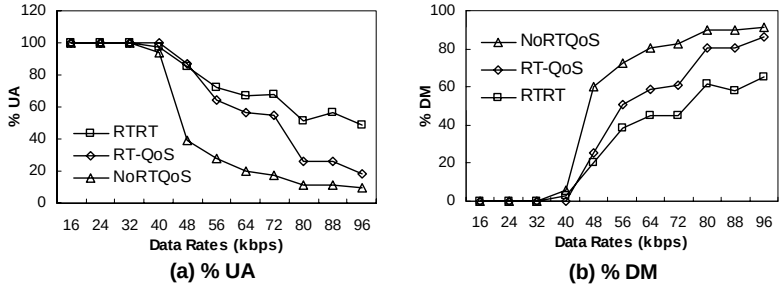
(a) % UA **(b) % DM**

Figure 2: Comparison of System-Wide %UA and %DM for Heterogeneous TUFs



(a) Step TUF (App3) **(b) Soft-Step TUF (App2)**

Figure 3: %UA for Different Applications



(a) % UA **(b) % DM**

Figure 4: Comparison of System-Wide %UA and %DM for Homogeneous TUFs