A testbed investigation of QoS mechanisms for supporting SLAs in IPv6

Vasilios A. Siris and Georgios I. Fotiadis
Dept. of Computer Science, Univ. of Crete and Inst. of Computer Science, FORTH
P.O. Box 1385, GR 711 10 Heraklion, Crete, Greece
Tel./fax: +30 2810 391726/391601, email: {vsiris,fotiadis}@csd.uoc.gr
November 2003

Abstract
We investigate the combined operation and interaction of QoS mechanisms, namely traffic policing, traffic shaping, and class-based queueing, in an IPv6 testbed, for TCP traffic. These mechanisms will be applied on edge routers at the interface between a user organization and his network provider, hence their tuning is important for effectively supporting Service Level Agreements (SLAs). Our results show how various parameters of the mechanisms affect the overall performance, hence can provide guidelines for appropriately selecting their values.

Keywords: policing, shaping, class-based queueing

1 Introduction
The IPv6 testbed consists of both Linux-based routers and Cisco 7X00 series routers running IOS Release 12.3, connected with Fast Ethernet links, Figure 1. For the Linux-based routers, traffic policing and shaping is performed by the Ingress Policer qldisc and the Token Bucket Filter mechanisms, and for the Cisco routers by the Traffic Policing and the Generic Traffic Shaping mechanisms. In this extended abstract we present the results only for Linux-based routers. The results for Cisco routers will appear in the full version of this paper. Note, however, that the qualitative conclusions in the case of Cisco routers are the same as those for Linux-based routers.

Both traffic policing and shaping are based on the token bucket algorithm, which characterizes the maximum amount of conforming traffic that a user can send. The token bucket has two parameters, a token rate \( R \) and a bucket size \( B \). A traffic shaper has one more parameter, the size \( Q \) of the buffer where packets are, if required, temporarily stored, waiting for a sufficient number of tokens to become available.

2 Policing
We first consider the case where only policing is applied. Figure 2 shows the aggregate throughput \( T \) as a function of bucket size \( B_p \), for different token rates \( R_p \) and number of TCP flows. As expected, the aggregate throughput increases with the bucket size, but always remains less than the policer’s rate \( R_p \). Indeed, for higher target rates the aggregate throughput is proportionally smaller: for \( R_p = 10 \) Mbps the maximum aggregate throughput is \( T = 9.7 \) Mbps, whereas for \( R_p = 50 \) Mbps the throughput is \( T = 32 \) Mbps and for \( R_p = 70 \) Mbps the throughput is \( T = 46 \) Mbps.

The dependence of the aggregate throughput on the bucket size exhibits a “knee”, i.e. for bucket sizes larger than some value, the rate of increase of the throughput is very small. Our results show that the bucket size where the “knee” appears is independent of the number of TCP flows, which agrees with [1]. Furthermore, the optimal bucket size \( B_p^* \) where the “knee” appears is higher for higher token rates. Indeed, our results suggest that for rates smaller than 40 Mbps, we have \( B_p^* \approx 0.6 \) Mbits + 0.03 sec \( \cdot R_p \), whereas for rates between 40 Mbps and 70 Mbps, we have \( B_p^* \approx 1.5 \) Mbits + 0.01 sec \( \cdot R_p \), hence the incremental increase of the bucket size for rates higher than 40 Mbps is smaller than for rates less than 40 Mbps. The work in [2], which considered rates up to 1.5 Mbps, suggests setting the bucket size to 0.5 sec \( \cdot R \), which gives a much higher value than the above equations. We conjecture that the difference is due to the higher rates that we consider and the different implementation of Cisco’s CAR (Committed Access Rate) mechanism, which was considered in [2].
3 Policing and Shaping

Next we consider the case where both policing and shaping is applied. Figure 3 shows the aggregate TCP throughput for different values of the shaper’s bucket size \(B_s\). First, observe that with the appropriate selection of \(B_s\), the aggregate throughput reaches 49.7 Mbps, whereas without shaping it was limited to 31.1 Mbps. Hence, shaping used in conjunction with policing can achieve throughput essentially equal to the policer’s token rate.

Regarding the dependence of the throughput on \(B_s\), we can make the following observations. First, for very small values of \(B_s\) the throughput is small; this can be attributed to the fact that small values of \(B_s\) limit the amount of traffic that can be sent. Second, for large values of \(B_s\), the aggregate throughput decreases to the value achieved when there is no shaping; this is because large values of \(B_s\) allow large bursts, hence diminishes the smoothing effect of shaping. Finally, observe that the number of TCP flows affects the range of values of \(B_s\) for which the maximum throughput is achieved. In particular, for a smaller number of flows the range is smaller. Indeed, observe that in the case of one flow, setting \(B_s = B_p\) yields a throughput smaller than the maximum.

The above results suggest that tuning of both the policing and the shaping mechanisms should be done in conjunction, taking into account the additional queueing delay introduced by shaping. Moreover, it would be interesting to investigate the effects of shaping for the case of token bucket marking [3].

4 Class-based Queueing

Our experimental results, that will appear in the full version of this paper, show that class-based queueing can be effectively used by both the user and the provider for giving different shares of capacity to different flows. Moreover, we have seen that shaping using Linux’s CBQ mechanism has a different behaviour in terms of the allowable bursts, compared to shaping using the Token Bucket Filter, hence its interaction with policing is different than that presented in the previous section.

Acknowledgements

This work was supported in part by GRNET (Greek Research and Technology Network), and performed within the EC funded project 6NET (IST-2001-32603).

References