Notes on CBQ and Guaranteed Service

Sally Floyd
Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, CA 94720
floyd@ee.lbl.gov

July 12, 1995

1 Introduction

This is a note on implementing guaranteed service using CBQ. We assume that a flow that has been accepted (by the admissions control procedure) for guaranteed service will be assigned its own priority-one class at the router. This note does not discuss the admissions control procedure.

2 The guaranteed service template

From the guaranteed service template, the router is given the TSpec (traffic specification) and RSpec (service specification) for the flow. The TSpec consists of a token bucket with a bucket depth b and a bucket rate r; the source should be policed at the edge of the network by this token bucket. The RSpec consists of a rate R, for $R \geq r$. The flow's service at the router is characterized by the bandwidth R and a buffer size B, where R represents the bandwidth that the flow is entitled to receive, and B represents the allocated buffer space in the router. It is assumed that the router will derive, from the TSpec, the required buffer space B to that no packets are dropped at the router.

The router exports two error terms C and D for a particular accepted flow. The router ensures that for a flow policed by the specified token bucket, the per-packet queueing delay at the router will be less than

$$b/R + C/R + D.$$

The service template gives the example of WRQ, where C is the MTU of the outbound link and D is 0. The queueing delay of b/R can be attributed to a bucket b of packets arriving instantaneously and effectively being transmitted at rate R. The C/R and D terms should include the queueing delay due to the transmission delay for the packet currently in service when a real-time packet arrives at the gateway, as well as the queueing delay due to other real-time packets scheduled to be sent before the packet in question.

3 CBQ

In CBQ, the priority-1 classes are served round-robin. The service discipline for the priority-1 classes can be either packet-by-packet round-robin (PRR) or weighted round-robin (WRR). For each class's turn in the round-robin scheduling, the amount of data that that packet gets to send depends on two factors, the scheduling algorithm (PRR or WRR), and whether or not that class is classified as overlimit. A class that is classified as overlimit might be additionally limited in the amount of data that it can send, being essentially rate-limited to its link-sharing bandwidth.

In CBQ, the worst-case delay for a packet from a guaranteed-service connection i occurs when packets from connection i arrive at the gateway just after connection i's turn in the round-robin scheme has passed.

Consider the delay for connection i at a particular gateway. Assume that there are at most k priority-1 classes, with the ith class having a bandwidth allocation of ρ_i. Let the capacity of the output link be ρ bytes/sec. (Let $p_i = \rho_i/\rho$ be connection i's allocated fraction of the bandwidth, for $\rho_i \geq R_i$, the entitled bandwidth for connection i. We assume that at most half of the bandwidth is allocated to priority-1 classes: $\sum_{i=1}^{k} p_i \leq 1/2$. (This is not because we believe that this is a necessary condition for providing guaranteed service in our scheme; this is simply because this greatly simplifies the analysis.)

3.1 Packet-by-packet round robin

The main reason to examine PRR is to explain why WRR scheduling gives better performance for guaranteed service traffic than WRR scheduling. With PRR

Draft

for servicing priority-1 classes, as in the current distribution of CBQ code, priority-1 classes are serviced in round-robin order, and each class that is not overlimit gets to send exactly one packet per round.

We assume that k is an upper bound on the possible number of priority-1 classes. Assume that all of the packets for the priority-1 classes contain at most $L = MTU$ bytes. Then a class-C_i packet that arrives at the gateway just after its "turn" has to wait at most $(k-1)L/\rho$ seconds for the other priority-1 classes to send a packet. (If a non-priority-one packet was in service when the class-C_i packet arrived, the class C_i also has to wait for the router to finish transmitting that packet. However, with CBQ no lower-priority packets will be sent when there is a non-overlimit higher-priority class with data to send.) If each packet for class C_i has at least s_i bytes, then the bandwidth received by class C_i in each round of the round-robin is at least

$$\frac{s_i}{(k-1)L + s_i}\rho.$$

If

$$\frac{s_i}{(k-1)L + s_i}\rho > R_i, \qquad (1)$$

where R_i is class C_i's entitled bandwidth, then class C_i receives at least its entitled bandwidth in every round-robin round. In this case, class-C_i's delay due to the wait for its turn in the round-robin is at most $(k-1)L/\rho$ seconds.

Equation 1 is likely to hold for a link where the aggregate bandwidth allocated to priority-one classes is small relative to the link bandwidth. Equation 1 is also likely to hold for a priority-one class that does not send small data packets and does not have a disproportionate allocation of the bandwidth among the priority-one classes.

If equation 1 does not hold, the packets arriving for class C_i at rate r could accumulate in a queue for that class until some of the other priority-one classes are classified as overlimit, and rate-limited to their allocated bandwidth. In this case, in order to calculate the worst-case delay for class C_i packets, we would have to calculate the worst-case time until other priority-one classes that are receiving more than their share of the bandwidth can be classified as overlimit, and then calculate the worst-case time until the accumulated backlog for class i has been dispersed. This depends on such factors as the time constant used by the estimator in estimating the bandwidth used by each class. Rather than going through these rather tedious calculations, we point out below that for a version of WRR scheduling where each priority-one class is guaranteed to receive its allocated bandwidth in each round of the round-robin, it is much more straightforward to bound the worst-case delay of an arriving guaranteed-service packet.

3.2 Weighted round robin

In our implementation of WRR, the priority-1 classes are served in round robin order. Each priority-1 class that is not overlimit is allowed to send $p_i M$ bytes. The parameter M is required to be at least $2kL$, for L the maximum packet size in bytes, and k the maximum number of priority-one classes that will be active at one time. Each class is allowed to finish sending the packet that contains the $p_i M$-th byte. If a class exceeds its allotment by x bytes, then its allotment for the next round is decremented by that amount. We say that a class that has exceeded its allotment in this fashion has a *deficit*.

Note that with this implementation of WRR, for a priority-1 class with a sufficiently large allocation (that is, where $p_i M$ is larger than the biggest packet for that class), that class will never be forced to "wait out" a round of the round robin due to a large accumulated deficit. However, given a static value for M, if none of the priority-1 classes with "larger" allocations have data to send, and all of the priority-1 classes with "smaller" allocations have to "wait out" a round of the round robin due to a large accumulated deficit, then the scheduling algorithm could (conceptually) take several rounds of the round robin before reducing the deficits and finding some priority-one class that is able to send a packet.

We explore the worst-case delay for a class-C_i packet that arrives at the gateway with no class-C_i backlog and no deficit.

Observation 1: At most

$$\sum_{i=1}^{k} (p_i M) + kL \leq M/2 + kL$$

bytes can be transmitted in one round of the priority-one round-robin. (Each class can send at most $L-1$ bytes over its allotted number of bytes in one round. This also uses the assumption that at most half of the link bandwidth is allocated to priority-1 classes.) □

Observation 2: Each priority-one class is guaranteed to receive its allocated share of the bandwidth in each round-robin round, unless some of this allocation was used 'in advance' in the most recently-sent packet from that connection.

Proof: Priority-1 class C_i gets to send at least $p_i M$ bytes in a round (unless some of those bytes were sent 'in advance' in the most recent packet). Thus class C_i gets a fraction at least

$$\frac{p_i M}{M/2 + kL} \geq p_i$$

of the bandwidth in each round-robin round. (This calculation uses the assumption that $M \geq 2kL$.) □

Draft

Observation 3: For a priority-one class with no backlog at the gateway and with no deficit, an arriving packet from class C_i can wait at most

$$\frac{M/2 + kL}{\rho} \geq \frac{2kL}{\rho}$$

seconds before receiving service. (This includes the transmission delay for the packet in service when the class-C_i packet arrives at the gateway.) □

Thus, a CBQ node that implements WRR for providing guaranteed service could advertise the constants

$$C = 0$$

and

$$D = \frac{M/2 + kL}{\rho}.$$

3.3 Comparisons of Weighted Round Robin and Weighted Fair Queueing

Note that unlike weighted fair queueing, with our implementation of weighted round robin the worst case packet delay is the same for any priority-1 class, and is not a function of the allocated bandwidth for that class. In comparison, the worst-case packet delay in WFQ, L/R, is a function of the bandwidth R assigned to that class.

The worst-case delay for high-bandwidth guaranteed service classes could be reduced further, if desired, by giving those classes more than one turn in each round of the round-robin. The frame structure is the order in which classes are listed for one round-robin round. 'Choosing a good frame structure' in Section 1.6.3 of Parekh's thesis gives an easy algorithm for creating a good frame structure. This kind of change would move WRR closer to a fluid-flow scheduling model.

An essentially equivalent way to explore the space between the WRR and WFQ scheduling algorithms would be to modify the version of WRR described in this paper to reduce M, the maximum number of bytes that can be sent in one round of the round robin without "borrowing" from rounds of the round-robin, and to place restrictions on the number of bytes that a class can borrow from future rounds. This kind of change would move WRR closer to a fluid-flow scheduling model: rounds would be smaller, a class with a large allocation would be guaranteed to receive its allocated bandwidth in every round of the round robin, and a class with a small allocation would be guaranteed to receive its allocated bandwidth in every k rounds of the round robin.

References

[F95] Floyd, S., "References on CBQ (Class-Based Queueing)", URL http://www-nrg.ee.lbl.gov/floyd/cbq.html.

[FJ95] Floyd, S., and Jacobson, V., "Link-sharing and Resource Management Models for Packet Networks." To appear in IEEE/ACM Transactions of Networking, August, 1995.

[P92] Parekh, A., "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks", Technical Report LIDS-TR-2089, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1992.

[SP95] Shenker, S., and Partridge, C., "Specification of Guaranteed Quality of Service", IEFT Internet Draft, draft-ietf-intserv-guaranteed-svc-01.txt, work in progress. July 1995.