Scheduling Algorithms

Jin Guang

January 10, 2022

Contents

1 Introduction

1.1 Setting

2 Static Priority

2.1 $M/M/1$ Queue with Priorities

3 Weighted Fair Queueing

We will first introduce weighted fair queueing (WFQ) by bit-by-bit round robin (BR), which is mainly from the paper by Demers, Keshav, and Shenker 1989. Alternatively, we will also introduce WFQ by another scheduling policy, called generalized processing sharing, which is mainly from the paper by Parekh and Gallager 1993.

3.1 From Bit-by-Bit Round Robin to Weighted Fair Queue

The main part follows the paper by Demers, Keshav, and Shenker 1989. We will first introduce the bit-by-bit round robin (BR) algorithm, which transmits data of each flow bit-by-bit to get the fair bandwidth. Motivated by BR, packet-by-packet fair queuing (FQ) algorithm selects transmission order for the packets by modeling the finish time for each packet as if they could be transmitted bit-by-bit round robin, and the packet with the earliest finish time according to this modeling is the next selected for transmission. Whereas FQ shares the link's capacity in equal subparts, weighted fair queueing (WFQ) allows schedulers to specify, for each flow, which fraction of the capacity will be given.

3.1.1 Bit-by-bit Round Robin

The bit-by-bit round robin algorithm has the following properties:

- The scheduler transmits the data bit-by-bit;
- This service discipline allocates bandwidth fairly since at every instant in time each flow is receiving its fair share.

Now, we introduce some notations to make it more clearly.

- $R(t)$: the number of rounds made in the round-robin service discipline up to time t, then $R(t)$ is a continuous function, with the fractional part indicating partially completed rounds.
- $N_{\text{ac}}(t)$: the number of active flow, i.e., those that have bits in their queue at time t.
- μ: the data rate of the outgoing line.

Then we will have following relations:

$$\frac{\partial R}{\partial t} = \frac{\mu}{N_{\text{ac}}(t)}.$$

If the line works in a constant rate μ and N_{ac} is unchanged during time $[t_1, t_2)$, the increment of the number of rounds will be

$$R(t_2) - R(t_1) = \frac{\mu}{N_{\text{ac}}(t_1)}(t_2 - t_1). \tag{3.1}$$

A packet of size P bits whose first bit gets serviced at time t_0 will have its last bit serviced P rounds later, at time t such that

$$R(t) = R(t_0) + P. \tag{3.2}$$

Now we define the time of the packet i belonging to flow α:

- ${t_i}^{\alpha}$: the time that the packet arrives at the gateway;
- ${S_i}^{\alpha}$ and ${F_i}^{\alpha}$: the values of $R(t)$ when the packet started and finished service;
- ${P_i}^{\alpha}$: the size of the packet.

Then the following relations hold:

$${F_i}^{\alpha} = {S_i}^{\alpha} + {P_i}^{\alpha} \text{ and } {S_i}^{\alpha} = \max\left({F_{i-1}}^{\alpha}, R\left({t_i}^{\alpha}\right)\right). \tag{3.3}$$

Therefore, the ordering of the ${F_i}^{\alpha}$ values is the same as the ordering of the finishing times of the various packets in the BR discipline. For this reason, we often abuse ${F_i}^{\alpha}$ as the finish time.

3.1.2 Packet-by-packet Fair Queueing Algorithm

Now we emulate this impractical algorithm by a practical packet-by-packet transmission scheme.

Given the functions $R(t)$ and $N_{\text{ac}}(t)$, the quantities ${S_i}^{\alpha}$ and ${F_i}^{\alpha}$ depend only on the packet arrival times ${t_i}^{\alpha}$ and not on the actual packet transmission times (see (3.1), (3.2) and (3.3)), as long as we define a flow to be active whenever non-idle. We are thus free to use these quantities in defining our packet-by-packet transmission algorithm.

A natural way to emulate the bit-by-bit round-robin algorithm is to let the quantities ${F_i}^{\alpha}$ define the sending order of the packets. The packet-by-packet transmission algorithm is simply defined by the rule:

- Non-preemptive: whenever a packet finishes transmission, the next packet sent is the one with the smallest value of ${F_i}^{\alpha}$.
- Preemptive: newly arriving packets whose finishing number ${F_i}^{\alpha}$ is smaller than that of the packet currently in transmission preempt the transmitting packet.

Although the actual quantity ${F_i}^{\alpha}$ for existing packets is potentially affected by new arrivals, ${F_i}^{\alpha}$ on the virtual time line is unchanged - the virtual time line warps with respect to real time to accommodate any new transmission.

For practical reasons, we have implemented the non-preemptive version, but the preemptive algorithm (with resumptive service) is more tractable analytically.

3.1.3 Weighted Fair Queueing

Whereas FQ shares the link's capacity in equal subparts, WFQ allows schedulers to specify, for each flow, which fraction of the capacity will be given.

Suppose a scheduler handling N flows is configured with one weight w_i for each flow. Then, the flow i will achieve an average data rate of $\frac{w_i}{w_1+\cdots+w_N}\mu$, where μ is the link rate. Here, we can imagine that the scheduler has $\sum w_j$ queues, and there are w_j queues to transmit the packet with the flow j.

The new rule corresponding to (3.1) is

$$R(t_2) - R(t_1) = \frac{\mu}{\sum_{j \in N_{\text{ac}}(t_1)} w_j}(t_2 - t_1), \tag{3.4}$$

where the total queues for active class are $\sum_{j \in N_{\text{ac}}} w_j$. And the new rule corresponding to (3.3) is

$$F_i^{\alpha} = S_i^{\alpha} + P_i^{\alpha}/w_i \text{ and } S_i^{\alpha} = \max\left(F_{i-1}^{\alpha}, R\left(t_i^{\alpha}\right)\right). \tag{3.5}$$

where there are w_i queues to transmit the packet with the flow i so we just need P_i^{α}/w_i rounds.

3.2 From Generalized Processing Sharing to Weighted Fair Queue

The main part follows the paper by Parekh and Gallager 1993. We will first introduce the generalized processing sharing (GPS) algorithm, which is a fluid and idealized model and cannot be used in practice. Similar as previous part, WFQ algorithm selects transmission order for the packets by modeling the finish time for each packet as if they could be transmitted by GPS, and the packet with the earliest finish time according to this modeling is the next selected for transmission.

3.2.1 Generalized Processing Sharing (GPS)

GPS is a fluid model where the queues are represented as reservoirs: customer arrivals increase the level of the reservoir by an amount of fluid corresponding to the service requirement; the reservoir leaks at a rate proportional to ϕ_i such that the overall output of K reservoirs is equal to the link rate r.

Let $W_k(t)$ be the workload of k-th queue at the time t, then

$$W_k(t) = W_k(0) + J_k(t) - \int_0^t \frac{\phi_k 1_{W_k(\tau)>0}}{\sum_{i=1}^K \phi_i 1_{W_i(\tau)>0}} r d\tau,$$

where $J_k(t)$ is the total workload arriving at time $[0, \tau)$.

3.2.2 Weighted Fair Queue (WFQ)

Let F_p be the time at which packet p will depart (finish service) under GPS. Now, suppose that the server becomes free at time t. Then WFQ server picks the first packet that would

complete service in the GPS simulation if no additional packets were to arrive after time t, i.e.,

$$p^* = \arg\min_{p \in S} F_p$$

where the set S includes all packets who stays in the system at time t.

3.2.3 Relation between WFQ and GPS

The difference of departure time of the same packet in the different system is bounded by the maximum packet length over service rate:

$$F_p^{WFQ} - F_p^{GPS} \leq \frac{L_{\max}}{r}.$$

The difference of workload process in the different system is also bounded by the maximum packet length:

$$Q^{WFQ}(t) - Q^{GPS}(t) \leq L_{\max}.$$

4 Deficit Round Robin

References

Demers, Alan, Srinivasan Keshav, and Scott Shenker (1989). "Analysis and simulation of a fair queueing algorithm". In: *ACM SIGCOMM Computer Communication Review* 19.4, pp. 1–12.

Parekh, Abhay K and Robert G Gallager (1993). "A generalized processor sharing approach to flow control in integrated services networks: the single-node case". In: *IEEE/ACM transactions on networking* 1.3, pp. 344–357.