Static vs accumulating priorities in healthcare queues under heavy loads

Binyamin Oz* Seva Shneer† Ilze Ziedins‡

Abstract

Amid unprecedented times caused by COVID-19, healthcare systems all over the world are strained to the limits of, or even beyond, capacity. A similar event is experienced by some healthcare systems regularly, due to for instance seasonal spikes in the number of patients. We model this as a queueing system in heavy traffic (where the arrival rate is approaching the service rate from below) or in overload (where the arrival rate exceeds the service rate). In both cases we assume that customers (patients) may have different priorities and we consider two popular service disciplines: static priorities and accumulating priorities. It has been shown that the latter allows for patients of all classes to be seen in a timely manner as long as the system is stable. We demonstrate however that if accumulating priorities are used in the heavy traffic or overload regime, then all patients, including those with the highest priority, will experience very long waiting times. If on the other hand static priorities are applied, then one can ensure that the highest- priority patients will be seen in a timely manner even in overloaded systems.

1 Introduction

We are currently seeing the effect COVID-19 has on healthcare services in vast majority of countries in the world. Healthcare services are also under increasing pressure as demographics change and populations age. Public health services are struggling, and sometimes failing, to maintain services under this increasing load. Given this context of high demand for tightly constrained resources it is instructive to reassess the rationales for prioritization regimes currently in use, and contrast with other possibilities. Specifically, in this paper we characterize the performance of the very commonly used static priority regime, and contrast it with the recently proposed accumulating priority regime, under critical loadings.

Healthcare systems have traditionally used static priority queues in a range of settings from triage in an emergency departments (EDs), to organizing access to elective surgeries, such as hip and knee replacements [5],[1]. In a static priority regime, patients are assigned to a priority class, and must wait to be treated until all patients in higher priority classes have been treated. Patients may sometimes be moved to a higher priority class if their condition deteriorates, but in practice there is often no automatic mechanism for making such transitions, and any adjustments may rely on patients proactively approaching their healthcare provider. Accumulating priority queues (APQs) have recently been proposed to overcome some of the inherent drawbacks of static priority queues in healthcare [9]. In accumulating priority queues, patients accumulate priority with time spent in the queue, at a rate that depends on their priority class, with higher priority patients accumulating priority faster than lower priority patients. Priority can accumulate linearly, or in a nonlinear fashion. Observational studies of behaviour in emergency departments have revealed that in practice physicians may operate a regime that is similar to an APQ, with the likelihood of being seen increasing more rapidly as waiting times approach threshold targets, see e.g. [5].

*Hebrew University of Jerusalem
†Heriot-Watt University
‡The University of Auckland

The accumulating priority regime was first proposed by Kleinrock [6], who obtained expressions for the expected waiting times for all classes. A large-deviations principle has been established in [11]. More recently, Stanford et al. [9] derived expressions for the Laplace Stieltjes Transform of the waiting time distribution, which can then be inverted numerically. A later paper [7], showed that a wide range of possible accumulation functions (including, for instance, exponential and log) have an equivalent linear regime, in the sense that the order in which patients are seen is the same in both the nonlinear and linear formulation.

This paper considers the performance of a single server queue with total arrival rate ρ and service rate 1, where ρ, the load on the server, either satisfies $\rho \uparrow 1$ or $\rho > 1$. The heavy traffic regime has been intensively studied, although not for the accumulating priority queue. When $\rho > 1$ queues are overloaded and hence unstable, and no equilibrium exists. Unstable queues, if unchecked, grow without bound, which is unrealistic for almost every application, and of course, an infinite queue never exists in practice. However, there are many applications where arrival rates are greater than service rates for shorter or longer periods. Traffic networks are an immediate example. In many cities rush hour queues build up, and then decay, not because the system can cope with the increased volume of traffic but simply because the flow into the network has reduced as rush hour passes.

At the time of writing, one cannot overstate the importance of looking at healthcare systems in the situations when demand suddenly exceeds capacity. We see healthcare systems of a very large number of countries being overwhelmed with an influx of patients.

Healthcare can however suffer from a similar excess of demand over capacity in other situations, particularly in winter, and for those healthcare systems which experience loss of capacity over longer periods, addressing the issue of how best to organize patient prioritization is vital [8]. Emergency departments may also see increased demand due to patients who could have been treated by their GP, but were deterred by cost (e.g. in New Zealand, where hospital visits in the public health system are free, but GP visits are not), or long waiting times for GP appointments. No patient arriving in a period when $\rho > 1$ actually sees an infinite queue – rather they see a large, possibly very large, queue ahead of them. Their waiting time for treatment, given they have joined the queue, is also not infinite, but as a patient's condition deteriorates, it may nevertheless be far too long. In practice also, if ρ is close to 1, then over relatively short periods of time it may be difficult to determine whether $\rho < 1$ or $\rho > 1$. Thus there is a strong practical need to address the question of how best to organize patient prioritization in this transient regime.

Indeed, we will see below that the two cases: a) $\rho << 1$ and b) $\rho \uparrow 1$ or $\rho > 1$, require fundamentally different approaches to patient prioritization. If $\rho << 1$, then the accumulating priority regime ensures that all patients are seen in a timely manner, while ensuring health targets are met (if those targets are feasible). On the other hand, if $\rho \uparrow 1$ or $\rho > 1$, then it is impossible to limit waiting times for all patients, and the static priority regime provides a mechanism for ensuring that healthcare is still available to the most acute, and vulnerable, patients (see Fig. 2 below), whereas under APQ the expected waiting times for all classes increase (Fig. 1 below). For this scenario we propose below a mixture of the two prioritization schemes.

Section 2 gives a detailed description of the models we consider. Section 3 considers the case where $\rho = 1 - \epsilon$, as $\epsilon \downarrow 0$, while Section 4 considers the case where $\rho > 1$. We conclude with a short discussion of related models and potential future research directions in Section 5.

2 Model

We consider a service system with a single server and $N+1$ classes of customers. Customers (patients) of class i arrive according to a Poisson process with rate λ_i and their priority class is associated with a positive real number b_i for $1 \leq i \leq N+1$. The higher the number b_i, the higher the priority class, and without loss of generality we assume $b_1 > b_2 > \ldots > b_N > b_{N+1}$. Thus arrivals of class 1 are in the highest priority class and arrivals of class $N+1$ are in the lowest priority class.

If a customer finds the server idle upon its arrival, the server immediately starts serving this customer. Priority is non-preemptive, that is, if the server is busy when a new customer

arrives, the customer joins the waiting room regardless of her priority class. The room is assumed to be of infinite size. Whenever the server finishes serving a customer and the waiting room is non-empty, the server starts to serve the customer with the highest current priority among those currently in the waiting room.

We consider two different priority policies:

- **Static priorities (SP)**: in this case a customer of class i has priority level b_i which does not change;
- **Accumulating priorities (AP)**: in this case a customer of class i that spent s, $s \geq 0$, units of time waiting in the waiting room has priority level $b_i s$.

We assume, without loss of generality, that service times for customers are independent and exponentially distributed with mean 1. Therefore

$$\rho = \lambda_1 + \ldots + \lambda_{N+1} \tag{1}$$

is the load on the system – the average amount of new work arriving per unit of time – and $\rho < 1$ is necessary for stability.

We will consider the system in high loads in two scenarios. In the first scenario we assume that $\rho = 1 - \varepsilon$ and $\varepsilon \downarrow 0$. The system is stable for any $\varepsilon > 0$, but, as ε decreases to 0, the average number of customers waiting in the queue in the stationary regime (and, by Little's formula, the average waiting time of a typical customer arriving in the system) tends to infinity. In the second scenario $\rho > 1$ and the system is thus unstable.

Our goal is to study the behavior of the expected waiting time of a customer from each of the classes, in the two scenarios described above, and under the two different priority policies.

3 Loads approaching capacity

In this section we consider a sequence of systems indexed by $\varepsilon > 0$ such that the loads in these systems increase to 1. More precisely, we assume that $\lambda_1(\varepsilon), \ldots, \lambda_{N+1}(\varepsilon)$ are non-decreasing functions of $\varepsilon > 0$ such that $\lambda_i(\varepsilon) \uparrow \lambda_i$ as $\varepsilon \downarrow 0$ for all $i = 1, \ldots, N+1$,

$$\sum_{i=1}^{N+1} \lambda_i(\varepsilon) = 1 - \varepsilon$$

for all $\varepsilon > 0$, and

$$\sum_{i=1}^{N+1} \lambda_i = 1.$$

An interesting special case of the setting above is the scenario where $\lambda_1, \ldots, \lambda_N$ are fixed and $\lambda_{N+1}(\varepsilon) = 1 - (\lambda_1 + \ldots + \lambda_N) - \varepsilon$, but we do not restrict ourselves to this case.

Regardless of the service discipline chosen, the systems are stable for any $\varepsilon > 0$. Denote by $(Q_1^\varepsilon, \ldots, Q_{N+1}^\varepsilon)$ the vector of steady-state queue lengths and by $(W_1^\varepsilon, \ldots, W_{N+1}^\varepsilon)$ the vector of steady-state waiting times (inclusive of the service time), for a particular value of ε. We also let $\mathbf{Q}_i^\varepsilon = \mathbb{E}(Q_i^\varepsilon)$ and $\mathbf{W}_i^\varepsilon = \mathbb{E}(W_i^\varepsilon)$ be the expected queue length and waiting times respectively, for all i. Regardless of the service discipline,

$$\mathbf{Q}_1^\varepsilon + \ldots + \mathbf{Q}_{N+1}^\varepsilon = \frac{1-\varepsilon}{\varepsilon} \sim \frac{1}{\varepsilon} \tag{2}$$

as $\varepsilon \to 0$. The total queue lengths thus increase to infinity, and we are interested in how queue lengths, and waiting times, of the individual classes behave.

3.1 Static priorities

Consider first the **SP** priorities. Let $\sigma_i(\varepsilon) = \sum_{k=1}^{i} \lambda_k(\varepsilon), 1 \leq i \leq N+1$, with $\sigma_0(\varepsilon) = 0$. Let also $\sigma_i = \lim_{\varepsilon \downarrow 0} \sigma_i(\varepsilon)$. From Cobham [4], we obtain that the expected waiting times for the priority classes are given by

$$\mathbf{W}_i^\varepsilon = \frac{1-\varepsilon}{(1-\sigma_{i-1}(\varepsilon))(1-\sigma_i(\varepsilon))} + 1, \qquad 1 \leq i \leq N+1. \tag{3}$$

and we can write

$$\mathbf{W}^{\varepsilon}_{N+1} = \frac{\frac{1}{\varepsilon}-1}{\varepsilon+\lambda_{N+1}(\varepsilon)}+1.$$

Thus, we see that as $\varepsilon \to 0$,

$$\begin{aligned} \mathbf{W}^{\varepsilon}_{i} &\to \frac{1}{(1-\sigma_{i-1})(1-\sigma_i)}+1<\infty, \qquad 1\le i\le N. \\ \mathbf{W}^{\varepsilon}_{N+1} &\to \infty \end{aligned}$$

and

$$\begin{aligned} \mathbf{Q}^{\varepsilon}_{i} &< \infty, \qquad 1\le i\le N \\ \mathbf{Q}^{\varepsilon}_{N+1} &\sim 1/\varepsilon. \end{aligned}$$

Thus, as $\varepsilon \to 0$, in the **(SP)** case, the expected queue lengths and waiting times for classes 1 to N are bounded from above, while for class $N+1$ they grow without bound. The durations of busy periods also increase without bound.

3.2 Accumulating priorities

For the **AP** case we can conclude from the Kleinrock formula ([6], see also [9]) that waiting times for all customers grow without bound as $\varepsilon \to 0$, so that if the b_i are held constant, this regime does not offer the same protection for the higher priority classes as **SP** does. Indeed we can prove the following exact statement.

Lemma 1. *Consider an accumulating priority queue with $N+1$ classes, and accumulation rates $b_1 > b_2 > \ldots > b_{N+1}$. Then*

$$\lim_{\varepsilon\downarrow 0} \varepsilon \mathbf{W}^{\varepsilon}_{i} = \frac{1/b_i}{\sum_{k=1}^{N+1} \lambda_k/b_k}, \qquad 1\le i\le N+1.$$

We present a proof of Lemma 1 below but first comment on its implications. The result may be interpreted as follows: a customer from class i entering service after waiting for time W_i has, at that time, priority $b_i W_i$. Lemma 1 essentially states that the **(AP)** discipline makes all these priorities just before service equal on average, across classes, in heavy traffic. This is similar to the behaviour in heavy traffic of the MaxWeight protocol (see [10]) which equates scaled queue lengths (we discuss this connection further below).

Lemma 1 also implies that $\mathbf{W}^{\varepsilon}_{i} \to \infty$ as $\varepsilon \downarrow 0$ for all i, and hence

$$\lim_{\varepsilon\downarrow 0} \varepsilon \mathbf{Q}^{\varepsilon}_{i} = \lim_{\varepsilon\downarrow 0} \varepsilon\lambda_i(\varepsilon) \mathbf{W}^{\varepsilon}_{i} = \frac{\lambda_i/b_i}{\sum_{k=1}^{N+1} \lambda_k/b_k}, \tag{4}$$

which also means that

$$\lim_{\varepsilon\downarrow 0} \frac{\mathbf{Q}^{\varepsilon}_{i}}{\sum_{k=1}^{N+1} \mathbf{Q}^{\varepsilon}_{k}} = \frac{\lambda_i/b_i}{\sum_{k=1}^{N+1} \lambda_k/b_k}. \tag{5}$$

Proof of Lemma 1. In order to simplify notation, in this proof we will drop the dependence of λ_i and $\mathbf{W}_i$ on ε. The formulas from [6] adapted to our setting are as follows:

$$\mathbf{W}_i = \frac{1/\varepsilon - 1 - \sum_{k=i+1}^{N+1} \lambda_k \left(1-\frac{b_k}{b_i}\right) \mathbf{W}_k}{1-\sum_{k=1}^{i} \lambda_k \left(1-\frac{b_i}{b_k}\right)}.$$

We use a proof by induction on $i = N+1, \ldots, 1$. First write

$$\begin{aligned} \mathbf{W}_{N+1} &= \frac{1/\varepsilon-1}{1-\sum_{k=1}^{N+1}\lambda_k\left(1-\frac{b_{N+1}}{b_k}\right)} = \frac{1/\varepsilon-1}{1-\sum_{k=1}^{N+1}\lambda_k + b_{N+1}\sum_{k=1}^{N+1}\lambda_k/b_k} \\ &= \frac{1/\varepsilon-1}{\varepsilon+b_{N+1}\sum_{i=1}^{N+1}\lambda_k/b_k}, \end{aligned}$$

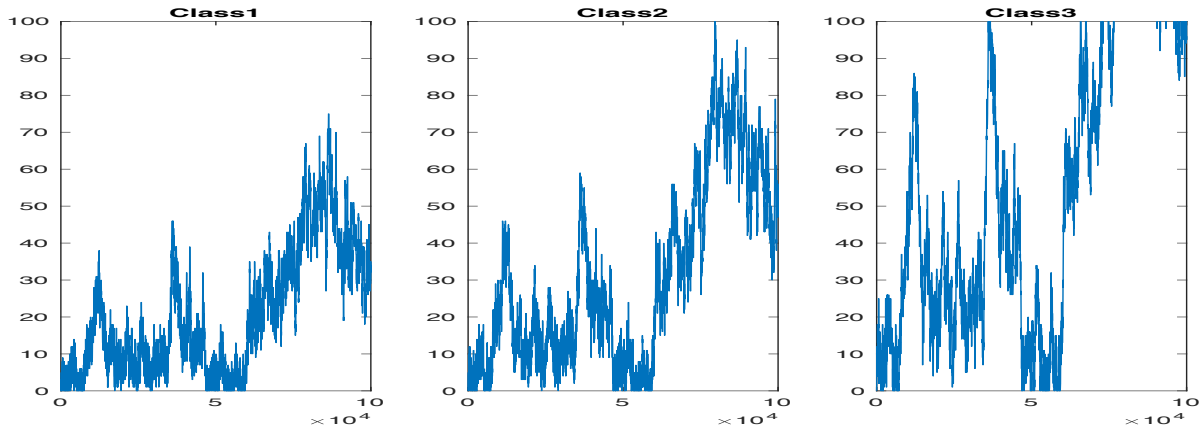

Figure 1: Numbers of customers of different classes against time in the system with accumulating priorities, with $\varepsilon = 0.001$

which implies the statement of the lemma for $i = N+1$. Assume now the statement is valid for all $i \geq j+1$ and let us prove if for $i = j$:

$$\begin{aligned}
\lim_{\varepsilon \downarrow 0} \varepsilon \mathbf{W}_j &= \frac{1}{1 - \sum_{k=1}^{j} \lambda_k \left(1 - \frac{b_j}{b_k}\right)} \left(1 - \sum_{k=j+1}^{N+1} \lambda_k \left(1 - \frac{b_k}{b_j}\right) \lim_{\varepsilon \downarrow 0} \varepsilon \mathbf{W}_k\right) \\
&= \frac{1}{1 - \sum_{k=1}^{j} \lambda_k + b_j \sum_{k=1}^{j} \lambda_k / b_k} \left(1 - \frac{\sum_{k=j+1}^{N+1} \lambda_k \left(1 - \frac{b_k}{b_j}\right) 1/b_k}{\sum_{k=1}^{N+1} \lambda_k / b_k}\right) \\
&= \frac{1}{\sum_{k=1}^{N+1} \lambda_k / b_k} \frac{1}{1 - \sum_{k=1}^{j} \lambda_k + b_j \sum_{k=1}^{j} \lambda_k / b_k} \left(\sum_{k=1}^{N+1} \lambda_k / b_k - \sum_{k=j+1}^{N+1} \lambda_k / b_k + \frac{1}{b_j} \sum_{k=j+1}^{N+1} \lambda_k\right) \\
&= \frac{1}{\sum_{k=1}^{N+1} \lambda_k / b_k} \frac{\sum_{k=1}^{j} \lambda_k / b_k + \frac{1}{b_j} \sum_{k=j+1}^{N+1} \lambda_k}{\sum_{k=j+1}^{N+1} \lambda_k + b_j \sum_{k=1}^{j} \lambda_k / b_k} = \frac{1/b_j}{\sum_{k=1}^{N+1} \lambda_k / b_k}.
\end{aligned}$$

□

3.3 Solution

These results suggest that in the **AP** regime, the accumulation rates need to be adjusted if the system is subjected to increasingly heavy loads. A natural solution we propose is to take $b_i(\varepsilon) = c_i/\varepsilon$ for fixed c_i, $\{1, \ldots, N\}$, with $b_{N+1} = c_{N+1}$. This effectively applies the **SP** regime to the lowest priority class, while maintaining the positive benefits of the **AP** regime for the remaining classes. More generally, we could consider regimes where $b_i(\varepsilon) = c_i/\varepsilon$ for fixed c_i, $\{1, \ldots, M\}$ for any $M \leq N$. If $M < N$, then the split between classes following **AP** and those following **SP** will still provide benefits to the higher priority classes. Whatever the split, only the lowest priority class will experience delays that grow without bound.

3.4 Numerical illustrations

We illustrate the conclusions above with typical sample paths of a system with $N = 2$ (there are therefore 3 classes of customers) under different priority regimes. Assume that arrival intensities are given by $\lambda_1 = \lambda_2 = 1/3$ and $\lambda_3 = 1/3 - \varepsilon$, and priorities $b_1 = 3$, $b_2 = 2$ and $b_3 = 1$.

In the case of accumulating priorities one can see (Fig. (1)) that the numbers of customers in all classes become large. If static priorities are used (Fig. (2)), only the number of customers of class 3 grows large, whereas the numbers of customers in classes 1 and 2 remain reasonably small at all times. We can also see from Fig. (3) below that if our

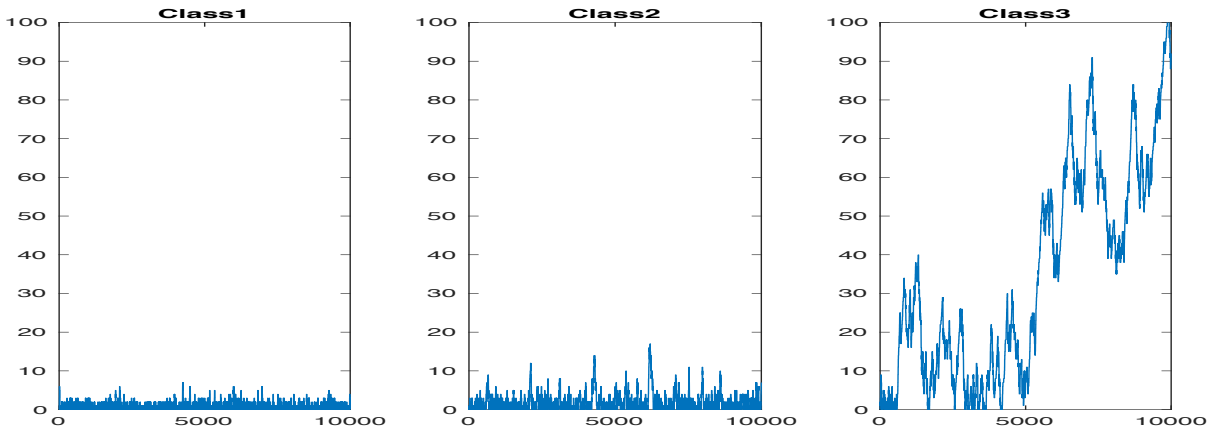

Figure 2: Numbers of customers of different classes against time in the system with static priorities, with $\varepsilon = 0.001$

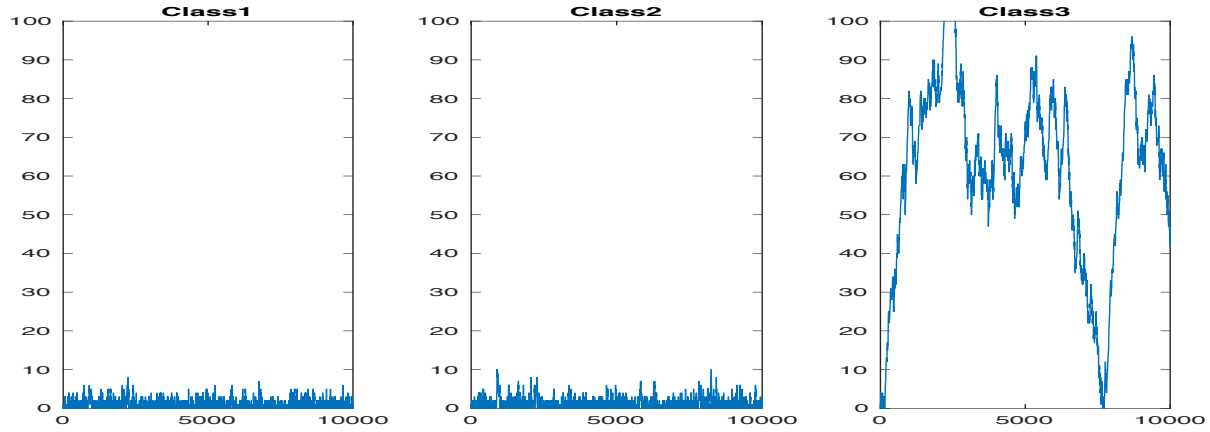

Figure 3: Numbers of customers of different classes against time in the system with accumulating but where initial priorities are assigned according to our suggested solution, with $\varepsilon = 0.001$

proposed solution is applied, the sample path looks similar to that of the system with static priorities.

A further example (see figure 4) illustrates how by changing the priority regime one can avoid long waiting times for high-priority customers if there is a sudden surge of lower-priority customers. There are, as before, three priority classes, $\lambda_1 = \lambda_2 = 1/3$ for the entire simulation. In the first quarter of the simulation time $\lambda_3 = 1/3 - 0.3$, so the total load of the system is 0.7, the system is in relatively light traffic and the accumulating priorities are used. All queue lengths are small. In the second quarter of the simulation time the arrival rate of the low priority customers suddenly jumps (this could represent for instance seasonal effects) to $\lambda_3 = 1/3 - 10^{-3}$, so the total load in the system is $1 - 10^{-3}$ and the system is in heavy traffic. One can observe all queues becoming large, including that of the highest priority customers. At this point we switch to the static priority regime and apply it for the remainder of the simulation time (arrival rates remain such that the system is in heavy traffic). One can observe that the static priority regime ensures that only the low-priority queue is large, queues of higher-priority customers are small.

4 Loads above capacity

In this section we assume that $\Lambda = \lambda_1 + \ldots + \lambda_N < 1$ is fixed. We assume in addition that $\rho = \Lambda + \lambda_{N+1} > 1$ and is also fixed, and further that $\rho - \lambda_i < 1$ for any i (i.e. the

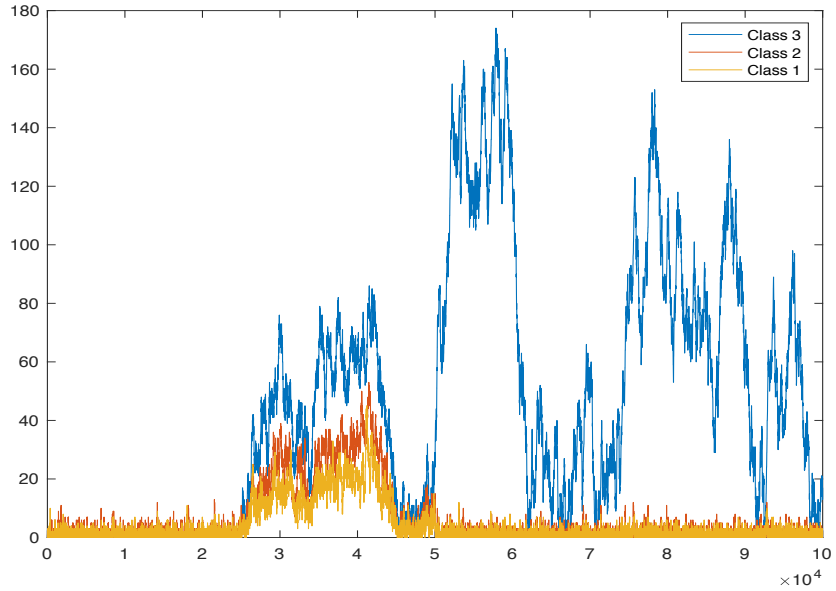

Figure 4: Switching from accumulating to static priorities stabilises queues of high-priority customers

system without any class would be stable). We have two particular cases in mind but do not restrict our attention to these. The first case is similar to the one we had in mind when studying the system in heavy traffic: an increase in the lowest-class arrival rate takes the system load above capacity. Another case may illustrate a catastrophic event, such as for instance a pandemic where a sudden jump in the highest-priority patients may lead to a system operating above capacity.

Since $\rho > 1$, and the system is therefore unstable, in this section we study a fluid version of the model in which we consider separately the queue for each customer class.

We suppose that the level of queue i at time $t > 0$, $L_i(t)$, is given by

$$L_i(t) = L_i(0) + \lambda_i t - \int_0^t D_i(s)ds,$$

where $D_i(s)$ denotes instantaneous service rate enjoyed by queue i.

4.1 Static priorities

In the case of **(SP)** $D_1(s) = \mathbf{I}(L_1(s) > 0) + \lambda_1 \mathbf{I}(L_1(s) = 0)$ for $s > 0$. As long as L_1, the queue for class 1 is strictly positive, all the available service capacity is directed to class 1. Once L_1 has emptied, the new arrivals of class 1 are assigned a dedicated service rate of λ_1. Since $\lambda_1 \leq \Lambda < 1$, this guarantees that $L_1(t) = 0$ for all $t \geq T_1$ for some finite T_1.

If $N \geq 2$, then $D_2(t) = 0$ for $t < T_1$. For values of $t > T_1$, we have $D_2(t) = (1 - \lambda_1)\mathbf{I}(L_2(t) > 0) + \lambda_2 \mathbf{I}(L_2(t) = 0)$, that is, a fraction λ_1 of the available rate is used to keep L_1 at zero, and the remaining service capacity is all assigned to queue 2, while it is positive. Once L_2 drops to 0, only a fraction λ_2 of the available capacity is required to drain the queue length at the same rate at which arrivals occur. Thus, since $\lambda_1 + \lambda_2 \leq \Lambda < 1$, $L_2(t) = 0$ for $t > T_2$ for some finite T_2.

Similarly to the above, we can conclude that there exists a finite $T = T_N$ such that $L_1(t) = \ldots = L_N(t) = 0$ for $t \geq T$. For class $N + 1$, on the other hand, $D_{N+1}(t) = 0$ for $t < T$. When $t \geq T$, $D_{N+1}(t) = (1-\Lambda)\mathbf{I}(L_{N+1}(t) > 0)$, and thus $L'_{N+1}(t) = \lambda_{N+1} - (1-\Lambda) = \rho - 1 > 0$, and $L_{N+1}(t) \to \infty$ as $t \to \infty$.

4.2 Accumulating priorities

When considering accumulating priorities we define the maximal priority process for each queue i, $1 \le i \le N+1$, as

$$P_i(t) = \frac{b_i L_i(t)}{\lambda_i}.$$

Here we have replaced $W_i(t)$ by $L_i(t)/\lambda_i$, which is the age of the oldest fluid particles in the system – for the fluid model considered here, these are equivalent.

If $\arg\max_j\{P_j(t)\}$ is unique, then

$$D_i(t) = \mathbf{I}(i = \arg\max_j\{P_j(t)\}), \qquad 1 \le i \le N+1.$$

On the other hand, if $\arg\max_j\{P_j(t)\}$ is not unique, let

$$\mathcal{J} = \{i : i \in \arg\max_j\{P_j(t)\}.$$

Under the accumulating priority regime, if two or more classes have priority $\max_j\{P_j(t)\}$ then service capacity should be divided between them in such a way that their priorities remain equal (and maximal). Therefore, if $|\mathcal{J}| > 1$, then $P_i'(t) = c$, say, for some constant $c \in \mathbb{R}_+$ for all $i \in \mathcal{J}$. Thus

$$c = P_i'(t) = \frac{b_i}{\lambda_i}(\lambda_i - D_i(t))$$

and hence

$$D_i(t) = \lambda_i - \frac{\lambda_i}{b_i}c$$

for all $i \in \mathcal{J}$. But $\sum_{i\in\mathcal{J}} D_i(t) = 1$ and hence

$$c = \frac{\sum_{i\in\mathcal{J}} \lambda_i - 1}{\sum_{i\in\mathcal{J}} \lambda_i/b_i}.$$

Recall that $c = P_i'(t)$ for all $i \in \mathcal{J}$, and recall that we assume $\rho - \lambda_i < 1$ for any i. Thus $P_i'(t) < 0$ for any $i \in \mathcal{J}$ as long as $\mathcal{J}$ does not consist of the entire set $\{1, \ldots, N+1\}$ and $P_i'(t) > 0$ as long as $\mathcal{J} = \{1, \ldots, N+1\}$.

Thus we can now understand the dynamics of the process of priorities: if we start at time 0 with a unique class with the highest fluid priority, its priority is decreasing until it equalises with the priority of another class. From that point onwards, the two priorities stay the same, and both are decreasing at the same rate, until they equalise with the priority of a further class. This continues until all priorities equalise, from which point onwards these priorities grow infinitely. This of course implies that the levels of fluids grow infinitely.

Note also that the above may be summarised for the level of queue i as follows: once all priorities have equalised,

$$L_i'(t) = \frac{\lambda_i}{b_i}P_i'(t) = \frac{\lambda_i}{b_i}c_i = (\rho - 1)\frac{\lambda_i/b_i}{\sum_{j=1}^{N+1} \lambda_j/b_j},$$

or

$$\lim_{t\to\infty} \frac{L_i(t)}{t} = (\rho - 1)\frac{\lambda_i/b_i}{\sum_{j=1}^{N+1} \lambda_j/b_j},$$

which shows that the relative queue lengths are exactly as in (5).

4.3 Solution

As before, a solution to the possibility of queues growing without bound if the accumulating priority regime is applied to all classes is to either employ a static priority regime, or a mixture of accumulating and static priority regimes, but in either case the lowest priority class needs to be operating under the static priority regime. Both the pure static priority regime, and the mixture, yield identical fluid solutions for classes $1, 2, \ldots, N$ as $t \to \infty$, with $L_i(t) = 0$, $t > T_S$ for some $T_S > 0$. On the other hand, $L_{N+1} \to \infty$ as $t \to \infty$, under any of the regimes.

5 Discussion

We have seen that in heavy traffic, the highest priority classes need greater protection than is afforded by the accumulating priority queue with fixed accumulation rates. This can be achieved either by permitting accumulation rates to grow in inverse proportion to ϵ in the case $\rho = 1 - \epsilon$, or by applying a static priority regime to the lowest priority class. In either case the lowest priority class suffers from increasing waiting times, but higher priority classes are protected from this growth.

These results have implications for other scenarios where prioritisation of tasks is a feature. We discuss below two other important areas of application, but we believe that the potential applications are considerably wider.

Prioritisation of tasks has been introduced in models of human dynamics where, upon completing a task, a person chooses the task from their to-do list with the highest priority to be performed next. A variant of static priorities has been considered in [2] and a version of accumulating priorities - in [3]. Few people would disagree with the observation that at least at some points in our lives we all experience an overload of our to-do lists. This may be modelled as the arrival rate being (perhaps temporarily) close to, or even above, the completion rate, exactly the settings considered in this paper. Our results can therefore be interpreted as follows: when the number of tasks on the to-do list grows, if time-dependent priorities are used, the number of outstanding high-importance tasks will grow. In order to prevent this, either static priorities, or a combination of accumulating and static priorities suggested here, should be used.

Another connection we would like to highlight is to wireless transmission protocols, namely the celebrated MaxWeight introduced in [12]. A simple version of it may be described as follows: there are a number of queues, each with its own exogenous stream of arriving jobs, and a single server which, upon completing a job, chooses the next one to perform from the queue with the largest number of outstanding jobs. Other priorities have also been discussed, in particular weighted queue lengths. If one views our model as tasks from the same class forming a queue, then in the case of accumulating priorities the server chooses the next task from the queue with the highest weighted waiting time of the longest-waiting task. Situations considered in this paper are such that the numbers of outstanding tasks in all queues grow to infinity. In this case, the waiting time of the longest-waiting customer is proportional to the number of outstanding tasks. Therefore, in the regimes considered here, the behaviour of the accumulating-priority queue is the same as that of the system governed by an appropriately weighted MaxWeight algorithm.

In this note we focused on average waiting times and queue lengths. It is of course important to study their distributions, which is a subject of our ongoing research. Another research direction we are currently pursuing is a more realistic scenario where customers abandon the system if they waited longer than a certain (perhaps random and perhaps class-dependent) threshold. Strategies minimizing the abandonment rate are of great practical interest.

References

[1] G. Arnett and D. Hadorn. Developing priority criteria for hip and knee replacement surgery: Results from the Western Canada waiting list project. *Canadian Journal of Surgery*, 46(4):290–296, 2003.

[2] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207, 2005.

[3] P. Blanchard and M.-O. Hongler. Modeling human activity in the spirit of Barabasi's queueing systems. *Physical Review E*, 75(2):026102, 2007.

[4] A. Cobham. Priority assignment in waiting line problems. *Journal of the Operations Research Society of America*, 2(1):70–76, 1954.

[5] Y. Ding, E. Park, M. Nagarajan, and E. Grafstein. Patient prioritization in emergency department triage systems: An empirical study of the Canadian Triage and Acuity Scale (CTAS). *Manufacturing and Service Operations Management*, pages 1–19, 2019.

[6] L. Kleinrock. A delay dependent queue discipline. *Naval Research Logistics Quarterly*, 11(3-4):329–341, 1964.

[7] N. Li, D. Stanford, P. Taylor, and I. Ziedins. Nonlinear accumulating priority queues with linear equivalent proxies. *Operations Research*, 65(6):1613–1628, 2017.

[8] L. Rolewicz and B. Palmer. The nhs workforce in numbers. 2019.

[9] D. A. Stanford, P. Taylor, and I. Ziedins. Waiting time distributions in the accumulating priority queue. *Queueing Systems*, 77(3):297–330, 2014.

[10] A. L. Stolyar et al. Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1):1–53, 2004.

[11] A. L. Stolyar and K. Ramanan. Largest weighted delay first scheduling: Large deviations and optimality. *Annals of Applied Probability*, pages 1–48, 2001.

[12] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *29th IEEE Conference on Decision and Control*, pages 2130–2132. IEEE, 1990.