

Journal Scientific and Applied Research, Vol. 31, 2026
International Journal

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

*Original Contribution*

# DUAL-QUEUE SCHEDULING AND AN ENGINEERING DELAY BOUND UNDER OVERLOAD

**Dragomir I. Vasilev[1], Srebrin V. Sharbanov[2]**

*[1]Department of Communication and Computer Engineering and Security Technologies, Faculty of Technical Sciences, Konstantin Preslavsky University of Shumen, Shumen 9712, 115 Universitetska St., e-mail: d.vasilev@shu.bg*
*[2]Department of Communication and Computer Engineering and Security Technologies, Konstantin Preslavsky University of Shumen, Shumen 9712, 115 Universitetska St., e-mail: srebrinsharbanov@gmail.com*

***Abstract:*** *One shared queue under overload is considered. Seven of every ten extractions are taken from the tail and three from the head. An engineering upper bound on the wait of jobs removed from the head is derived for a sustained hybrid and a constant share $\theta$. With eight servers, mean service 40 ms and a full buffer of 1000 jobs, the bound is about 16.7 s and sits above a 3 s threshold. A 10 Hz guard is therefore included in the sizing: it replaces LIFO with the hybrid when the head has waited more than 3 s. A measured P99 is not reported.*

***Key words:*** *Dual-Queue, Delay bound, FIFO, LIFO, Starvation, One deque.*

## 1. Introduction

Under overload, FIFO keeps new requests behind older ones. Pure non-preemptive LIFO takes the newest waiting request and does not interrupt a job already in service, so a request at the bottom may remain without a practical upper bound until the arrival rate falls [1].

This paper considers one queue with three extraction modes: from the head only, from the tail only, or seven times from the tail and three from the head. The third mode is called the hybrid. The aim is an engineering upper bound on the wait of requests that the hybrid removes from the head. Given the number of servers $\boldsymbol{c}$, the per-server service rate $\boldsymbol{\mu}$, the share $\boldsymbol{\theta}$, and a threshold $\boldsymbol{T_{\text{starve}}}$, the bound yields the count $\boldsymbol{N^*}$ up to which the hybrid alone meets the threshold, and shows when a buffer of size $\boldsymbol{K}$ is too deep for that threshold.

## 2. Problem setting

The setting is as follows. One queue of capacity $K$. There are $c$ servers, each with mean service $1/\mu$, so the pool capacity is $c\mu$. Arrivals have intensity $\lambda$. When $\lambda > c\mu$ and the queue is already full, a new request does not enter: it is a rejection. Rejections are counted separately from the wait of accepted requests. The bound below applies only to requests already inside.

What is sought is the following. For chosen $\boldsymbol{\theta}$ and threshold $\boldsymbol{T_{\text{starve}}}$, obtain the count $\boldsymbol{N^*}$ up to which extractions from the head alone meet the threshold, and check whether the given $\boldsymbol{K}$ lies above that count. Above it, the hybrid alone does not hold the threshold.

There is one queue. Every accepted request enters at the tail. FIFO removes from the head and preserves arrival order among those removals. LIFO removes from the tail. The hybrid does not split requests at entry and does not maintain two physical queues.

Of every ten successive extractions, seven are from the tail and three from the head. The share toward the head is $\boldsymbol{\theta = 0.30}$, and toward the tail about $\boldsymbol{0.70}$. The three head extractions are called the head channel below: a share of capacity, not a separate queue. Order among those head removals remains arrival order.

A ten-step scheme is used because the share is constant and easy to count. Random sampling with probability $\boldsymbol{\theta}$ has the same mean share and larger short-horizon variance. The bound below uses the seven-and-three scheme.

Suppose that during the ten extractions no new request arrives and twelve requests wait, ordered from head to tail: the oldest is first, the newest is twelfth. The first seven extractions take from the tail the twelfth, eleventh, tenth, ninth, eighth, seventh and sixth. The eighth, ninth and tenth steps take from the head the first, second and third. The fourth and fifth remain. In one cycle the seven newest and the three oldest leave. The middle two stay and wait for the next cycle, which again starts from the tail.

If new requests arrive during the cycle, they enter at the tail. They are not placed ahead of those already near the head. While a request sits near the tail, it is among the candidates for the seven tail extractions. If newer ones push it inward until it reaches the head, from there it waits for the three head extractions out of ten. The bound below is for that second stretch: how many head extractions still separate it from a server. The time it spent near the tail is not included in the formula.

In De Schepper, Albisser, Tilmans and Briscoe, Dual Queue is a pair of queues for L4S and classic TCP that share capacity [2]. Here Dual-Queue is the same seven-and-three rule on one queue.

At any moment only one of the three modes is active. FIFO removes every waiting request from the head. LIFO removes every waiting request from the tail. The hybrid alternates seven from the tail and three from the head. Which mode is in force is decided by a controller from the queue: its length, the share of completions below a freshness threshold, and the wait of the oldest request. That controller is called the agent. The bound does not describe how the agent reaches its choice. It holds while the choice remains the hybrid. If the agent has chosen LIFO and the oldest request has waited more than $\boldsymbol{3\ \text{s}}$, a separate check replaces the choice with the hybrid. That check is called the guard.

### 3. Bound on waiting from the head

Let the buffer hold $N$ requests, with $c$ servers each of rate $\mu$. Sustained overload is considered, $\lambda > c\mu$, so the queue is full and the hybrid stays on. If capacity is split with a constant share $\theta$, the head-channel rate is

$$\boldsymbol{r_{\text{FIFO}} = \theta \cdot c \cdot \mu} \tag{1}$$

For a request that waits for $N$ extractions on that channel, the conservative bound is

$$\boldsymbol{W_J \leq \frac{N}{r_{\text{FIFO}}} = \frac{N}{\theta \cdot c \cdot \mu}} \tag{2}$$

If instead of the full buffer one takes position $k$ on the head channel, $1 \le k \le N$,

$$W_{(k)} \le \frac{k}{\theta \cdot c \cdot \mu} \tag{3}$$

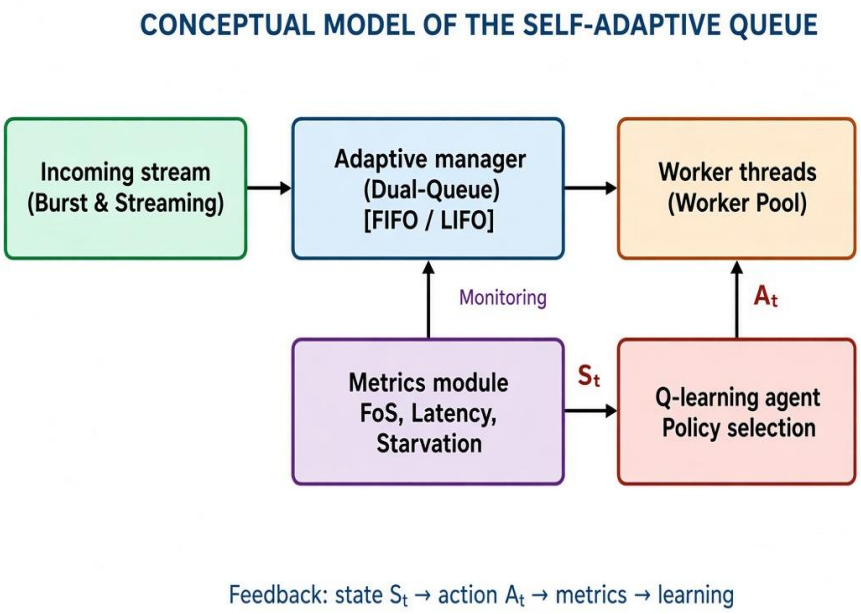



**Fig. 1.** One queue with extraction from both ends

The longest case in a full buffer remains $k = N$.

The formula comes from the mean head-channel speed. One server completes on average $\boldsymbol{\mu}$ requests per second. With $\boldsymbol{c}$ busy servers the pool completes $\boldsymbol{c\mu}$ requests per second. The hybrid assigns share $\boldsymbol{\theta}$ of extractions to the head, so the head advances at $\boldsymbol{\theta c\mu}$ requests per second. If $\boldsymbol{k}$ such extractions remain ahead of a request, the time to it is $\boldsymbol{k}$ divided by that speed. The numerator is a count of requests, the denominator is requests per second, and the quotient is in seconds. At $\boldsymbol{k = N}$ one obtains the time for the head channel to drain a full buffer. That is not the wait of the oldest request: it sits at the head. With the model numbers, $\boldsymbol{c\mu = 200}$ req/s and $\boldsymbol{\theta = 0.30}$ give $\mathbf{60}$ req/s toward the head. Every entry in the table is a division by that speed, or by the speed at another $\boldsymbol{\theta}$.

There are three assumptions. The hybrid is active for a sustained interval, not only in isolated intervals. The share $\boldsymbol{\theta}$ is nearly constant: the ten-step counter is shared by the pool so the ratio does not break across threads. Outside the bound remain the discrete step of the scheme and the dispersion of service time about its mean. The bound divides the extraction count by the mean rate and treats every service as $\boldsymbol{1/\mu}$.

The numerical example uses the model parameters: $\boldsymbol{c = 8}$ servers and mean service $\mathbf{40\ ms}$, so $\boldsymbol{\mu \approx 25\ s^{-1}}$ and total capacity $\boldsymbol{c\mu = 200}$ req/s. At $\boldsymbol{\theta = 0.30}$ the head receives

$$\boldsymbol{r_{\mathrm{FIFO}} = 0.30 \cdot 8 \cdot 25 = 60\ \text{req/s}} \tag{4}$$

The remaining about 140 req/s are taken from the tail. For threshold $T_{\mathrm{starve}} = 3$ s the count that the hybrid alone covers from the head is

$$\boldsymbol{N^{*} = T_{\mathrm{starve}} \cdot \theta \cdot c \cdot \mu = 3 \cdot 60 = 180} \tag{5}$$

Position $k = 50$ gives $50/60 \approx 0.83$ s. Position $k = 180$ gives 3.0 s. Position $k = 500$ gives 8.33 s. A full buffer $k = 1000$ gives 16.67 s. The same arithmetic at $N = 180$ and $\theta = 0.20$ gives 4.50 s, and at $\theta = 0.40$ gives 2.25 s. A smaller share toward the head lengthens the bound. A larger share shortens it and assigns more capacity to older requests, so the hybrid moves closer to FIFO.

**Table 1.** Values from the bound $W = N/(\theta c\mu)$ at the model parameters. The rows are not measurements.

| $N$ | $\theta$ | $c$ | $\mu$ | $W_{\max}$ **[s]** |
|---|---|---|---|---|
| 100 | 0.30 | 8 | 25 | 1.67 |
| 180 | 0.30 | 8 | 25 | 3.00 |
| 500 | 0.30 | 8 | 25 | 8.33 |
| 1000 | 0.30 | 8 | 25 | 16.67 |
| 180 | 0.20 | 8 | 25 | 4.50 |
| 180 | 0.40 | 8 | 25 | 2.25 |

At $\boldsymbol{K = 1000}$ the bound stays above the **3 s** threshold. The hybrid alone does not hold that threshold for a full buffer.

The model uses two different thresholds. The threshold below which a completed request counts toward freshness is **1 s**. The guard threshold is **3 s**. The bound is computed for the chosen threshold. At the same **60** req/s toward the head, a **1 s** threshold gives $\boldsymbol{N^* = 60}$, and a **3 s** threshold gives $\boldsymbol{N^* = 180}$. The hybrid alone keeps under **1 s** those requests with at most **60** head extractions remaining, and under **3 s** those with at most **180**. A buffer of **1000** exceeds both counts. For either threshold the formula alone does not cover a full buffer, so the guard remains in the sizing. This arithmetic is not a measured freshness share. It states how many head extractions fit into the threshold at a constant mean rate.

The choice of $\boldsymbol{\theta}$ is read from the same table. At $\boldsymbol{N = 180}$ and a target threshold of **3 s**, share $\boldsymbol{0.30}$ gives $\boldsymbol{3.00}$ **s** and sits on the threshold. Share $\boldsymbol{0.20}$ gives $\boldsymbol{4.50}$ **s** and exceeds it. Share $\boldsymbol{0.40}$ gives $\boldsymbol{2.25}$ **s** and stays under it. A larger share toward the head shortens the bound because more capacity removes older requests. A smaller share lengthens the bound and leaves more extractions for the tail, where the new ones sit. The bound does not price that second cost. It gives only the wait on the head channel. The operating point $\boldsymbol{\theta = 0.30}$ is the one at which $\boldsymbol{N = 180}$ coincides with the guard threshold.

The bound is not the mean wait of all requests. FIFO, LIFO and the hybrid may complete the same accepted requests on the same servers. The formula measures only the time to extraction on the head channel. A request taken from the tail does not pass through this account. After extraction comes service itself, mean **40 ms**. It sits after the bound and is not included in it. The guard threshold compares with queue wait, up to the timestamp of the oldest request, before it is extracted.

Position $\boldsymbol{k}$ counts how many head extractions remain until the request leaves through that channel. The request at the head has $\boldsymbol{k = 1}$: the next head extraction takes it. At **60** req/s that is about $\boldsymbol{0.017}$ **s**, plus waiting for its next turn in the ten-step cycle, up to **35 ms**. A request with $\boldsymbol{k}$ head extractions ahead waits, by the bound, $\boldsymbol{k/60}$ seconds. The conservative extreme $\boldsymbol{k = N}$ is a request for which a whole full buffer still stands on the head channel. At

$\boldsymbol{N = 1000}$ that is $\mathbf{16.67\ s}$: the time for the head channel to remove $\mathbf{1000}$ requests if the hybrid does not stop. A request near the tail usually leaves earlier through the seven tail extractions. The bound does not rely on them. Hence at $\boldsymbol{k = N}$ it is wide. The seven tail extractions remove newer requests and do not reduce $\boldsymbol{k}$: $\boldsymbol{k}$ falls only when the head is served. While the queue is not full, $\boldsymbol{k}$ is not equal to capacity $\boldsymbol{K}$. If forty head extractions currently stand ahead of a request, the formula uses $\boldsymbol{k = 40}$.

The request at the head, with $\boldsymbol{k = 1}$, waits by the formula about $\mathbf{0.017\ s}$ only while the hybrid is on and its turn in the ten-step cycle arrives. If the choice is pure LIFO, nothing is taken from the head, $\boldsymbol{k}$ does not move, and the head timestamp keeps growing. So a small $\boldsymbol{k}$ does not cancel the guard: the guard protects the head precisely when the hybrid is not chosen.

**Table 2.** Position $k$ and the bound at $\theta = 0.30$. The rows are divisions, not measurements.

| $k$ | **What stands ahead on the head channel** | $W$ **at** 60 **req/s** |
|---|---|---|
| 1 | at the head; next head extraction | 0.017 s |
| 50 | fifty head extractions | 0.83 s |
| 180 | the count $N^*$ for a 3 s threshold | 3.00 s |
| 1000 | a full buffer through the head channel only | 16.67 s |

Let a request in a full buffer of $\mathbf{180}$ have $\boldsymbol{k = 180}$ and let the hybrid stay on. The head channel removes $\mathbf{60}$ requests per second, so its remainder falls as follows:

**Table 3.** Remainder for a request with $k = 180$ under a continuous hybrid and 60 req/s toward the head. A second spent on pure LIFO does not drop a row of this table.

| $t$ **[s]** | **Removed from the head** | **Remaining ahead** | **Bound on the remainder [s]** |
|---|---|---|---|
| 0 | 0 | 180 | 3.00 |
| 1 | 60 | 120 | 2.00 |
| 2 | 120 | 60 | 1.00 |
| 3 | 180 | 0 | 0 |

The same reading covers a small buffer. At $\boldsymbol{N = 100}$ and $\boldsymbol{\theta = 0.30}$ the bound is $\mathbf{1.67\ s}$. That buffer sits under the $\mathbf{3\ s}$ threshold and above the $\mathbf{1\ s}$ threshold. The hybrid alone covers three seconds for one hundred requests from the head and does not cover one second for the same hundred. Which threshold is kept is chosen before the arithmetic. The formula then says whether $\boldsymbol{K}$ lies under the matching $\boldsymbol{N^*}$.

### 4. When the bound ceases to describe waiting

The bound describes waiting only while the choice is the hybrid and the share $\theta$ holds. If in a given interval the agent has chosen FIFO, every request is taken from the head and the seven-and-three formula does not describe that interval. If the choice is pure LIFO, there is no head channel and requests at the head receive no extraction.

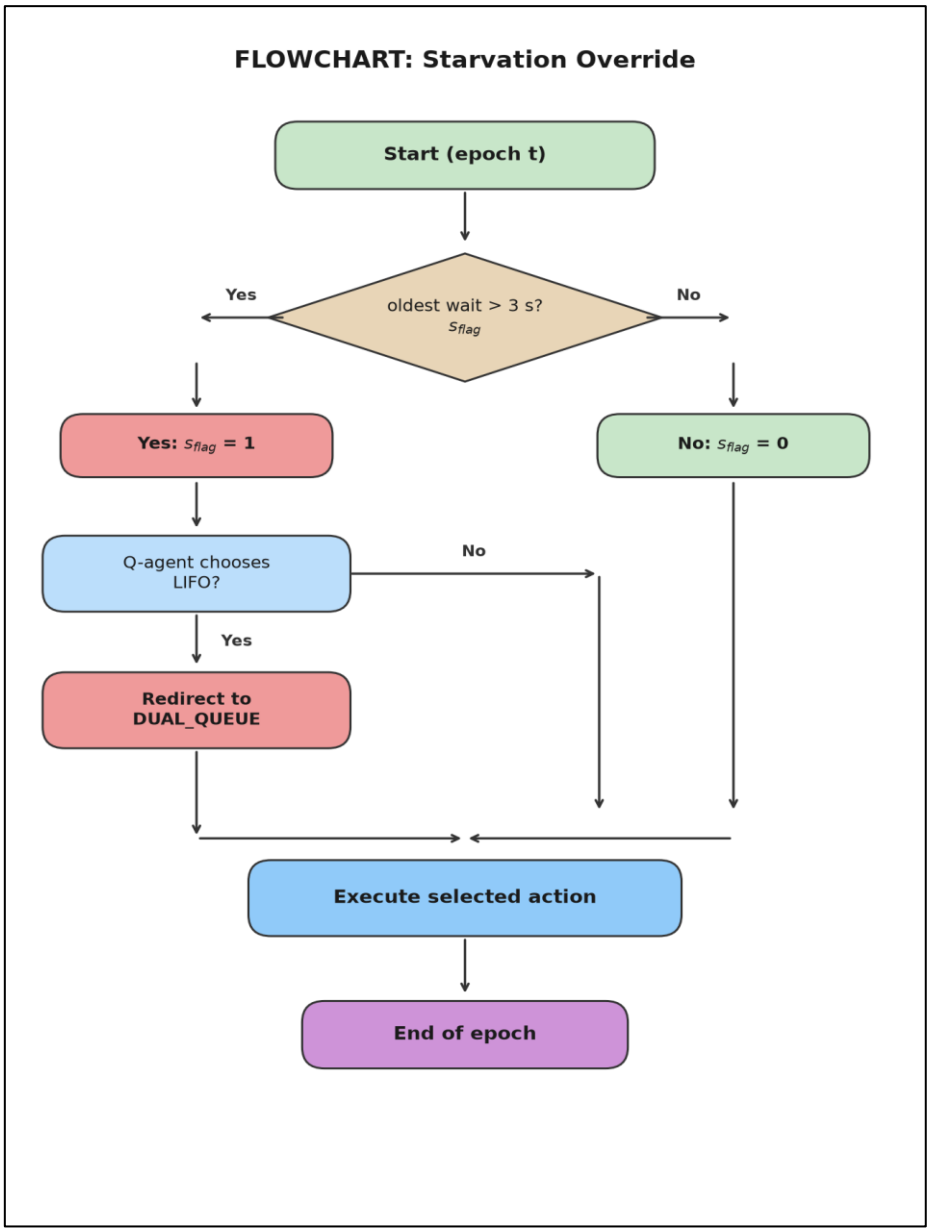



**Fig. 2.** If the head has waited more than 3 s and the action is LIFO, it is replaced by the hybrid

The scheme is a step, not a continuous flow. In ten successive extractions the head waits through the first seven, which go to the tail, and moves on the eighth, ninth and tenth. The formula replaces that step by a mean intensity of $\mathbf{60}$ req/s.

At capacity $\mathbf{200}$ req/s, ten extractions last $\mathbf{10/200 = 0.05\ s}$, that is $\mathbf{50\ ms}$. Of those, the seven toward the tail take $\mathbf{35\ ms}$, and the three toward the head take $\mathbf{15\ ms}$. Inside the cycle the head may wait up to those $\mathbf{35\ ms}$ before its extraction. Against a $\mathbf{3\ s}$ threshold that step is small. The formula smooths it to $\mathbf{60}$ req/s and does not write it separately.

Service-time dispersion also stays outside. At a mean of $\mathbf{40\ ms}$ a single request may finish much faster or much slower. The instant of the $\boldsymbol{k}$-th head extraction fluctuates about $\boldsymbol{k/60}$. The bound gives the upper account by mean intensity and does not describe that fluctuation.

Below capacity the queue is not full. Then the formula uses the current position $\boldsymbol{k}$ on the head channel. Capacity $\boldsymbol{K}$ is only the buffer ceiling and does not replace $\boldsymbol{k}$.

The number $\mathbf{16.67\ s}$ is the time to remove $\mathbf{1000}$ requests from the head at a continuous $\mathbf{60}$ req/s. It applies to a request with $\boldsymbol{k = 1000}$, and only if the hybrid stays on through the whole interval. The request at the head does not wait those $\mathbf{16.67\ s}$: it has $\boldsymbol{k = 1}$. If overload lasts two seconds and then arrivals drop, a request with large $\boldsymbol{k}$ also does not wait out the full-buffer account, because the queue ceases to be full and one thousand extractions no longer stand ahead of it. The bound for $\boldsymbol{K = 1000}$ is a ceiling under long overload and a continuous hybrid, not the wait at every short burst.

Between two guard checks the choice may be pure LIFO. The check runs every $\mathbf{100\ ms}$. On a saturated pool of capacity $\mathbf{200}$ req/s about $\mathbf{20}$ services complete in that interval. If they are taken from the tail, the head does not move and the bound time during that interval does not reduce its wait. So the bound is not a guarantee for an arbitrary action choice between two checks.

**5. Guard and sizing**

The guard runs at 10 Hz while the agent holds one of the three modes. Every 100 ms the timestamp of the oldest request is read. Let $s_{\text{flag}} = 1$ when it has waited more than $3000\ \text{ms}$, and let $a_{\text{chosen}}$ be the agent's choice. If that choice is LIFO and $s_{\text{flag}} = 1$, execution follows the hybrid and about 30% of capacity returns to the head. Otherwise the agent's choice is executed:

$$\boldsymbol{a^{\text{final}} = \begin{cases} \text{DUAL} \\ a_{\text{chosen}} \end{cases} \quad \begin{matrix} \\ \text{otherwise} \end{matrix}} \tag{6}$$

$$\boldsymbol{s_{\text{flag}} = 1 \wedge a_{\text{chosen}} = \text{LIFO}} \tag{7}$$

Here DUAL means the hybrid, seven from the tail and three from the head. The guard does not recompute the bound. It looks at the head wait and the current choice.

The $\mathbf{3000\ ms}$ threshold coincides with the bound at $\boldsymbol{N = 180}$. At moderate fill the hybrid alone reaches the $\mathbf{3\ s}$ threshold. At a full buffer of $\mathbf{1000}$ the bound is $\mathbf{16.67\ s}$ and the guard enters the sizing before waiting follows that account to the end. Because of the $\mathbf{100\ ms}$ step and service dispersion, a zero probability of exceeding the threshold is not claimed. A share of violations is not measured. A measured P99 is not reported.

The two thresholds and three capacities are placed side by side. At $\boldsymbol{\theta = 0.30}$ a $\mathbf{1\ s}$ threshold gives $\boldsymbol{N^* = 60}$, and a $\mathbf{3\ s}$ threshold gives $\boldsymbol{N^* = 180}$.

**Table 4.** Whether the hybrid alone covers the threshold. On a row ``above $N^*$'' the sizing includes a guard. The table is the same division, not an experiment.

| $K$ | **Relative to $N^* = 60$ for a 1 s threshold** | **Relative to $N^* = 180$ for a 3 s threshold** |
|---|---|---|
| 100 | above $N^*$; the formula does not cover the threshold | below $N^*$; the formula covers the threshold |
| 180 | above $N^*$ | on the threshold, $W = 3.00$ s |
| 1000 | above $N^*$ | above $N^*$, $W = 16.67$ s |

Sizing follows the bound.

1. The target threshold is taken as $T_{\text{starve}}$. In the example it is 3 s.

2. $c$ and $\mu$ are taken from the pool. Here $c = 8$ and $\mu \approx 25\,\mathrm{s}^{-1}$.
3. $\theta$ is chosen and $N^* = T_{\mathrm{starve}}\theta c\mu$ is computed. At $\theta = 0.30$ one obtains $N^* = 180$. If mean service is not $40\,\mathrm{ms}$, the rate changes. At $\mu = 20\,\mathrm{s}^{-1}$ and the same eight servers, capacity is $160$ req/s, the head channel is $48$ req/s, and for a $3$ s threshold one obtains $N^* = 144$. Table 1 holds for $\mu = 25\,\mathrm{s}^{-1}$. Other hardware is recomputed with the same formula.
4. If capacity $K$ is much above $N^*$, a guard is planned. For $K = 1000$ that is the case: $1000 \gg 180$.
5. A check under bursty load remains a separate experiment.
6. Together with $N^*$ the assumptions are recorded: a sustained hybrid, nearly constant $\theta$, and a guard at $10$ Hz.

## 6. Conclusion

Dual-Queue in this work is one queue with seven extractions from the tail and three from the head. The bound $W_{(k)} \leq k/(\theta c\mu)$ gives the wait on the head channel under a sustained hybrid and a constant share $\theta$. At $c = 8$, $\mu \approx 25\,\mathrm{s}^{-1}$ and $\theta = 0.30$, a $3$ s threshold corresponds to $N^* = 180$, and a full buffer of $1000$ yields about $16.7$ s. Therefore, for a deep buffer the sizing includes a $10$ Hz guard that replaces LIFO with the hybrid when the head has waited more than $3$ s. The bound is not a measured P99 and does not replace the guard.

## References


[1] Harchol-Balter, M. Performance Modeling and Design of Computer Systems: Queueing Theory in Action. Cambridge University Press, 2013.

[2] De Schepper, K., Albisser, O., Tilmans, O. & Briscoe, B. Dual queue coupled AQM: Deployable very low queuing delay for all. arXiv:2209.01078, 2022.