HAL
archives-ouvertes.fr

Performance analysis of space-time priority queues

Christian Carballo, Urtzi Ayesta, Dieter Fiems

▶ **To cite this version:**

Christian Carballo, Urtzi Ayesta, Dieter Fiems. Performance analysis of space-time priority queues. Performance Evaluation, Elsevier, 2019, 133, pp.25-42. 10.1016/j.peva.2019.04.003 . hal-02351580

HAL Id: hal-02351580
https://hal.archives-ouvertes.fr/hal-02351580

Submitted on 23 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: https://oatao.univ-toulouse.fr/26217

Official URL:

https://doi.org/10.1016/j.peva.2019.04.003

To cite this version:

Carballo, Christian and Ayesta, Urtzi and Fiems, Dieter *Performance analysis of space-time priority queues.* (2019) Performance Evaluation, 133. 25-42. ISSN 0166-5316.

Any correspondence concerning this service should be sent to the repository administrator: tech-oatao@listes-diff.inp-toulouse.fr

Performance analysis of space–time priority queues☆

C. Carballo-Lozano [b,e,*], U. Ayesta [a,d,f,g], D. Fiems [c]

[a] CNRS, IRIT, 2 rue C. Camichel, 31071 Toulouse, France
[b] DeustoTech - Fundación Deusto, Avda. de las Universidades 24, 48007 Bilbao, Spain
[c] Ghent University, Department of Telecommunications and Information Processing, 9000 Gent, Belgium
[d] IKERBASQUE - Basque Foundation for Science, 48011 Bilbao, Spain
[e] Universidad de Deusto, Facultad de Ingeniería, Avda. de las Universidades 24, 48007 Bilbao, Spain
[f] Université de Toulouse, INP, 31071 Toulouse, France
[g] UPV/EHU, University of the Basque Country, 20018 Donostia, Spain

A B S T R A C T

Depending on the application level quality requirements, packets on the Internet can broadly be classified into two classes: Streaming (S) and Elastic (E). Streaming packets require low delays, but can tolerate packet losses without degrading the user level performance. Elastic traffic has less stringent delay demands, but its performance is very sensitive to packet losses, since the latter trigger costly packet recovery mechanisms. In this paper, we analyze a space–time priority queue (STPQ), a queueing discipline that can satisfy the performance criterion of both classes. Streaming packets get time priority, i.e., they are served with priority over elastic packets, while elastic packets get space priority, i.e., in the event the buffer is full, the arrival of an elastic packet will trigger, with probability α, the push-out from the queue of a streaming packet. In our main analytical result, we have developed efficient algorithms to compute the steady-state probabilities and the mean delay of STPQ. We have also analyzed the queue in the light-traffic regime, which allows us to derive a closed-form approximation. Numerical results illustrate that the STPQ can provide a low delay to streaming packets without significant degradation of the blocking probability of elastic packets.

Keywords:
Priority queue
Performance analysis
Finite buffer
Blocking probability

1. Introduction

Based on the quality-of-service requirements at the application level, packets in the Internet can be broadly classified into two categories: Streaming and Elastic. Streaming packets correspond to audio, video and in general applications for which the main performance criterion is low delay and low delay jitter. In the event of congestion and packet losses, lost packets are not retransmitted and the server will vary the bit-rate in order to alleviate the congestion [1]. In contrast, elastic packets correspond to the transfer of digital documents (Web pages, files, ...). These transfers can tolerate variations of the delay at the packet level, but packet losses can be harmful as they often increase the delay at the application level. Indeed, in the event of a packet loss, the packet will be retransmitted and the source will reduce the sending rate [2]. In terms of transport layer protocols, we can associate streaming traffic to UDP, and elastic traffic to TCP. According to the Cooperative Association for Internet Data Analysis (CAIDA), TCP traffic represents 90% of the traffic, while UDP makes up

☆ Research partially supported by the French "Agence Nationale de la Recherche (ANR)" through the project ANR-15-CE25-0004 (ANR jCJC RACON).
* Corresponding author at: DeustoTech - Fundación Deusto, Avda. de las Universidades 24, 48007 Bilbao, Spain.
E-mail address: christian.carballo@deusto.es (C. Carballo-Lozano).

https://doi.org/10.1016/j.peva.2019.04.003

for the remaining 10% [3]. Similar traffic mixes are also reported by WIDE [4], a Japanese project that collects and publicly distributes measurement data on the web (see Section 7.2) for more details).

In order to accommodate this dichotomy in the traffic in the Internet, this paper studies a two class queueing system with a priority service discipline that we label as "space–time priority queue". Unlike traditional priority models, in our model one class gets priority in terms of service (time priority), while the other class is given priority to enter the queue in case the buffer is full (space priority). The time priority class is the natural class for streaming packets, while the space priority class is better suited for elastic packets.

The idea of treating elastic and streaming packets differently is not new, and it was for instance the starting point of the "alternate best-effort" [5], where the authors proposed a queue management algorithm that has resemblances to our "space–time" proposal. A very similar scheduling service was more recently proposed in [6]. However, to the best of our knowledge, literature lacks the analysis of a stochastic model to investigate the performance of the space–time queue. For this reason, this paper sets out to analyze the distribution of the number of packets and the mean waiting time in the space–time priority queue in steady-state. As not all packets receive service, we reserve the term waiting time for packets that receive successful service in the remainder, and not for packets that are pushed out. In other words, the expected waiting time is the time a packet spends in the queue, conditioned on receiving service.

The main contributions of our work are the following:

- We propose a new queueing model that we label as the space–time priority queue.
- We develop an efficient algorithm to calculate the steady-state probabilities, and a recursion to calculate the mean waiting time of both classes.
- In the numerical section, we show that the STPQ can provide a low delay to streaming packets without a significant degradation of the blocking probability of elastic packets.

The rest of the paper is organized as follows. In Section 2 we refer to the most important literature related to our work. Section 3 formally describes the STPQ model and introduces the performance metrics we consider. Section 4 presents the algorithm to efficiently calculate the steady-state probabilities and the mean waiting times. In Section 5 we carry out a light-traffic analysis of the queue. Section 6 presents the results from numeric computations. In Section 7 we present simulation results of STPQ with deterministic packet sizes and with a real trace taken from WIDE [4]. Finally, in Section 8 we derive some conclusions of the STPQ and our work.

2. Related work

Priority queueing is textbook material, covered for example in classical books by Kleinrock [7] and Cohen [8], as well as in the monographs by Jaiswal [9] and Takagi [10]. The classic setting most often assumes an infinite capacity buffer. Priority queues with finite buffers have received considerably less attention. For such buffers, time priority regulates the order in which the customers are served while space priority regulates buffer access. Space priority can be implemented in different ways.

The simplest way is the absence of space priority. In this case, customers enter the priority queue as long as there is buffer space. Buffer space is either shared [11] or each class gets its separate queue [12]. In partial buffer sharing, customers with space priority can enter the buffer as long as there is room, while customers without space priority do not enter once the buffer content exceeds a fixed threshold, see e.g. [13] where different types of buffer sharing strategies are compared. The combination of partial buffer sharing and time priority is studied in [14] for a discrete-time queue with batch-Markovian arrivals where the priority class gets both time and space priority and in [15] where one class gets time priority while the other gets space priority. A similar system is studied in [16] for continuous-time queues with phase-type service times, again giving space and time priority to different classes.

For the space-priority disciplines above, once a customer enters the queue, it remains there until it receives service. This is not the case for push-out buffers. Here, space-priority customers can push out other customers if the buffer is full upon arrival. Several authors have studied push-out buffers in the absence of time priority. Cheng and Akyildiz [17] and Lee et al. [18] consider a push-out buffer with Poisson arrival streams and general service time distributions, while Kapadia et al. [19] analyze the multi-server push-out buffer.

Some authors also analyze queueing models that combine time-priority and push-out. Lee and Choi consider a discrete-time (ATM) buffer with priority and push-out [20]. Each time-priority class gets its separate finite capacity buffer, which implements push-out. In [21] and [22], Avrachenkov et al. analyze a two-class priority queue, where, the class that gets time-priority also receives space-priority. In their model, if the buffer is full, the high priority queue pushes out with probability α a low priority packet. The authors use a generating function approach to develop an algorithm to solve the system of steady state equations. For the special case $\alpha = 1$ (so the strict space-priority case), [22] derives analytic equations for the blocking probabilities.

In contrast to existing literature and motivated by the requirements of streaming and elastic traffic, the present paper investigates the push-out priority buffer, where space and time-priority are given to different classes. To the best of our knowledge, all the existing literature deals with the strict priority case in which one class has space and time priority over the other. We note at this point that the analysis of the space–time priority queue is more complicated, because none of the classes has a strict priority over the other one.

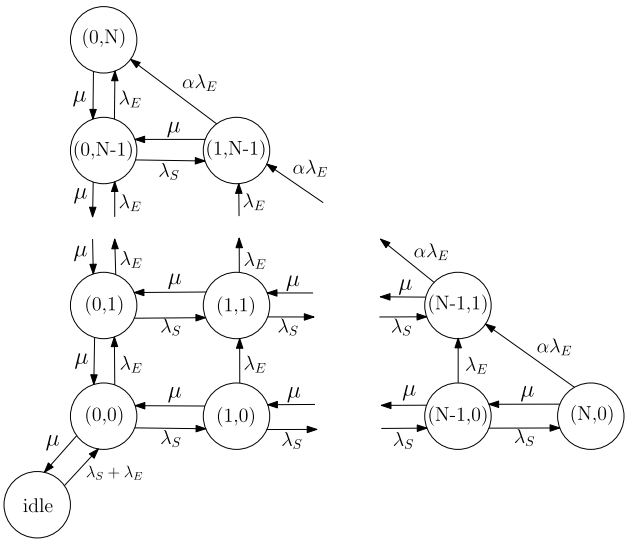


Fig. 1. State transition rate diagram for the STPQ.

3. Model description and preliminary results

We model the space–time priority queue as a non-preemptive priority queue with two classes of packets. Class-S (streaming) and class-E (elastic) packets arrive to the system according to a Poisson process of rate λ_S and λ_E, respectively. We denote by f_X the density function of a continuous random variable X. We assume that the service time distribution of both type of packets, denoted by the random variable B, is exponentially distributed with parameter μ, i.e., $f_B(x) = \mu e^{-\mu x}$. In particular, we have $\mathbb{E}(B) = 1/\mu$. The buffer has finite size N and it is shared by both classes of packets. Thus, the total number of packets that can simultaneously be in the system is $N + 1$, that is, N in the buffer and one in service. We let $L_E(t)$ and $L_S(t)$ denote the number of class-E and class-S packets in the buffer at time t, respectively.

Class-S (streaming) packets have time priority over class E (elastic) packets. This implies that a class-E packet gets served only if $L_S(t) = 0$. If the buffer is not full, that is $L_S(t) + L_E(t) < N$, an incoming packet from either class is accepted into the buffer. Class-E packets have space priority over class-S packets. Thus, if the buffer is full, an incoming class-S packet will not be accepted into the buffer, and an incoming class-E packet will push-out from the queue the last class-S packet with probability $0 \leq \alpha \leq 1$. The case $\alpha = 1$ is denoted as "non-randomized push-out", and it will be particularly present in our analysis. As we will see in the ensuing, the case $\alpha < 1$ is less amenable to analysis.

Let $p_{i,j} = \lim_{t\to\infty} \mathbb{P}(L_S(t) = i, L_E(t) = j)$ denote the steady-state probability that there are $i + j$ packets in the queue, out of which i are of type class-S and j are of type class-E. We also denote by $p_{\text{idle}} = \mathbb{P}(\text{idle})$ the stationary probability that the server is idle. Under the just described modeling assumptions, $(L_S(t), L_E(t))_{t\geq 0}$ is a Markov Chain, and the transition diagram characterizing the steady-state is depicted in Fig. 1, which expands vertically with class-E packets and horizontally with class-S ones for at most N total packets. Note that downward transitions only happen in the first column, when no class-S packets are waiting and push-out transitions can only happen in the boundary $L_S(t) + L_E(t) = N$.

Then, the steady-state probabilities are the unique solution of the following system of equations:

$$
\begin{aligned}
(\lambda_S + \lambda_E)p_{\text{idle}} &= \mu p_{0,0}\,, \\
(\lambda_S + \lambda_E + \mu)p_{0,0} &= \mu p_{1,0} + \mu p_{0,1} + (\lambda_S + \lambda_E)p_{\text{idle}}\,, \\
\mu p_{0,N} &= \lambda_E p_{0,N-1} + \alpha\lambda_E p_{1,N-1}\,, \\
(\alpha\lambda_E + \mu)p_{N,0} &= \lambda_S p_{N-1,0}\,, \\
(\lambda_S + \lambda_E + \mu)p_{0,j} &= \mu p_{1,j} + \mu p_{0,j+1} + \lambda_E p_{0,j-1}\,, && \text{for } 0 < j < N\,, \\
(\lambda_S + \lambda_E + \mu)p_{i,0} &= \mu p_{i+1,0} + \lambda_S p_{i-1,0}\,, && \text{for } 0 < i < N\,, \\
(\lambda_S + \lambda_E + \mu)p_{i,j} &= \mu p_{i+1,j} + \lambda_S p_{i-1,j} + \lambda_E p_{i,j-1}\,, && \text{for } 0 < i + j < N\,, \\
(\alpha\lambda_E + \mu)p_{i,j} &= \lambda_S p_{i-1,j} + \lambda_E p_{i,j-1} + \alpha\lambda_E p_{i+1,j-1}\,, && \text{for } i + j = N\,.
\end{aligned}
$$

We let $p_N = \sum_{i=0}^{N} p_{i,N-i}$ denote the probability that there are N packets in total in the queue. The traffic intensities for each class are denoted by $\rho_S = \lambda_S/\mu$ and $\rho_E = \lambda_E/\mu$, and thus $\rho = \rho_S + \rho_E$ is the total intensity. It is easy to see that

$$
p_N = \frac{1-\rho}{1-\rho^{N+2}}\rho^{N+1}\,,
$$

as the total number of packets in the buffer evolves as the buffer content of an $M/M/1/N{+}1$ queue with arrival rate $\lambda_S + \lambda_E$ and service rate μ. Similarly, we note that the total probability along diagonals can be directly retrieved from known formulas of the $M/M/1/N{+}1$ queue. Thus, the probability of having k packets is then

$$\sum_{i=0}^{k} p_{i,k-i} = \frac{(1-\rho)\rho^{k+1}}{1-\rho^{N+2}}\,, \tag{1}$$

while the idle and empty buffer probabilities equals,

$$p_{\text{idle}} = \frac{(1-\rho)}{1-\rho^{N+2}}\,, \quad p_{0,0} = \frac{(1-\rho)\rho}{1-\rho^{N+2}}.$$

The main performance criteria we are interested in are the loss probabilities for both class-S, denoted by $p_{\text{loss}}^{(S)}$, and class-E, $p_{\text{loss}}^{(E)}$, as well as the mean waiting times for both classes, denoted by, $\mathbb{E}[W_S]$ and $\mathbb{E}[W_E]$. These performance measures are determined in the following sections.

4. Main results

In this section we state our main results regarding the performance criteria steady state probabilities, packet loss probabilities and mean waiting times for class-S and class-E packets.

4.1. Steady-state probabilities

The steady-state probabilities can be found by solving the system of balance equations. Since the Markov chain is finite and irreducible, the balance equations have a unique normalized solution. Computing the steady-state probabilities using matrix inversion can be computationally heavy when N grows. Exploiting the structure of the Markov chain, see Fig. 1, we however can calculate these probabilities more efficiently. The key for our approach is the observation that we can solve the steady state probabilities per row because: *(i)* downward transitions are only possible in the first column, i.e., in states with no class-S packets and *(ii)* we know the probability mass on the diagonals. The next proposition describes the algorithm to calculate the probabilities.

Proposition 1. *The following algorithm finds the unique solution of the system of steady state equations in $O(N^2)$ steps:*

1. *Probabilities $p_{i,0}$:*
   (a) *Set $b_N = 1$ and $b_{N-1} = \frac{\alpha\lambda_E+\mu}{\lambda_S}$ and recursively calculate*
   $$b_i = b_{i+1}\frac{\lambda_E+\lambda_S+\mu}{\lambda_S} - b_{i+2}\frac{\mu}{\lambda_S}$$
   for $i = N-2,\ldots,0$.
   (b) *Given the values b_i, calculate*
   $$p_{i,0} = \frac{b_i}{b_0}\frac{(1-\rho)\rho}{1-\rho^{N+2}}\,,$$
   for $i = 0,\ldots,N$.
2. *Probabilities $p_{i,j}$, for $j = 1,\ldots,N$:*
   (a) *Set $a_{N-j}^{(j)} = 0$ and $a_{N-j-1}^{(j)} = -(p_{N-j,j-1} + \alpha p_{N-j+1,j-1})\frac{\lambda_E}{\lambda_S}$, and recursively calculate for $i = N-j-2,\ldots,0$,*
   $$a_i^{(j)} = a_{i+1}^{(j)}\frac{\lambda_E+\lambda_S+\mu}{\lambda_S} - a_{i+2}^{(j)}\frac{\mu}{\lambda_S} - p_{i+1,j-1}\frac{\lambda_E}{\lambda_S}\,,$$
   (b) *Calculate $p_{0,j}$,*
   $$p_{0,j} = \frac{(1-\rho)\rho^{j+1}}{1-\rho^{N+2}} - \sum_{i=1}^{j} p_{i,j-i}\,,$$
   and for $i = 1,\ldots,N-j$, calculate,
   $$p_{i,j} = a_i^{(j)} + b_{i+j}\frac{p_{0,j}-a_0^{(j)}}{b_j}\,.$$

Proof. The proof can be found in Appendix A. □

Remark 1. Proposition 1 can be easily extended to the case of state-dependent push-out probabilities, that is, when α_i, $i = 1, \ldots, N$, depends on i, the number of class-S packets waiting. The latter can be of interest to reduce the push-out probability when the number of class-S packets is small.

Carefully counting the number of operations, we find that the computational complexity of Algorithm 1 is $O(N^2)$. As the number of states is $O(N^2)$ one cannot do better if one wants to calculate all probabilities. We note however that it may be possible to find faster algorithms which are $O(N^2)$ as well.

Algorithm 1 Efficient algorithm for steady-state probabilities.

procedure STEADYSTATE($N, \alpha, \lambda_S, \lambda_E, \mu$)
  $b(N) \leftarrow 1$
  $b(N-1) \leftarrow \frac{(\alpha\lambda_E+\mu)}{\lambda_S}$
  for $i = N-2, \ldots, 0$ **do**
    $b(i) \leftarrow b(i+1)\frac{\lambda_E+\lambda_S+\mu}{\lambda_S} - b(i+2)\frac{\mu}{\lambda_S}$
  end for
  for $i = 0, \ldots, N$ **do**
    $p(i,0) \leftarrow \frac{b(i)}{b(0)}\frac{(1-\rho)\rho}{1-\rho^{N+2}}$
  end for
  for $j = 1, \ldots, N$ **do**
    $a(N-j) \leftarrow 0$
    $a(N-j-1) \leftarrow -(p(N-j, j-1) + \alpha p(N-j+1, j-1))\frac{\lambda_E}{\lambda_S}$
    for $i = N-j-2, \ldots, 0$ **do**
      $a(i) \leftarrow a(i+1)\frac{\lambda_E+\lambda_S+\mu}{\lambda_S} - a(i+2)\frac{\mu}{\lambda_S} - p(i+1, j-1)\frac{\lambda_E}{\lambda_S}$
    end for
    $p(0,j) \leftarrow \frac{(1-\rho)\rho^{j+1}}{1-\rho^{N+2}} - \sum_{i=1}^{j} p(i, j-i)$
    for $i = 1, \ldots, N-j$ **do**
      $p(i,j) = a(i) + b(i+j)\frac{p(0,j)-a(0)}{b(j)}$
    end for
  end for
  return p
end procedure

4.2. Loss probabilities

The expression for the loss probabilities is given in the following proposition.

Proposition 2. *The loss probabilities of streaming and elastic packets are given by the following formulae*

$$p_{loss}^{(E)} = p_{0,N} + (1-\alpha)(p_N - p_{0,N}), \tag{2}$$

$$p_{loss}^{(S)} = p_N + \alpha\frac{\rho_E}{\rho_S}(p_N - p_{0,N}), \tag{3}$$

where

$$p_N = \frac{1-\rho}{1-\rho^{N+2}}\rho^{N+1}.$$

Proof. An arriving elastic packet at time t will be lost either *(i)* when the buffer is full of elastic packets, $L_E(t) = N$, or *(ii)* with probability $1-\alpha$ in case the buffer is full ($L_S(t) + L_E(t) = N$) and the number of streaming packets satisfies $0 < L_S(t) \leq N$. The probabilities of those events are $p_{0,N}$ and $\sum_{i=1}^{N} p_{i,N-i}$, respectively. Thus, by the PASTA property (Poisson Arrivals See Time Averages) we obtain (2).

The stream of class-S packets lost is formed by two streams: those lost when the buffer is full, which happens at rate $\lambda_S p_N$, and those lost by being pushed-out by class-E packets, which happens at rate $\alpha\lambda_E(p_N - p_{0,N})$. Summing these two rates and equating with $\lambda_S p_{\text{loss}}^{(S)}$, yields (3). □

4.3. Waiting time

In this section we assume that the steady-state probabilities $p_{m,n}$ are known (see Proposition 1) and we show how to calculate the mean waiting time of both classes. The waiting time of interest is the one which takes into account only the packets that are actually served.

For the elastic traffic, since a job that enters the queue is never pushed out, we can apply Little's Law to the number of elastic customers in the system. We note that in this case the effective arrival rate is $\lambda_E(1 - p_{\text{loss}}^{(E)})$. Thus, we conclude

that

$$\mathbb{E}[W_E|\text{served}] = \frac{\mathbb{E}[L_E]}{\lambda_E\left(1-p_{\text{loss}}^{(E)}\right)}. \tag{4}$$

The waiting time for the class-S packets is considerably more complicated since a packet that enters the queue can be lost later on. We now consider a tagged class-S packet waiting in the queue and define the following mean waiting time, conditioned on the number of class-E packets and the number of class-S packets before and after the tagged packet in the queue,

$$w(m, m', n) = \mathbb{E}[W_S 1_{\{\text{served}\}}|m \text{ class-}S \text{ before, } m' \text{ class-}S \text{ after, } n \text{ class-}E].$$

By the PASTA property, the probability that an arriving packet finds $L_S(t) = m$ class-S and $L_E(t) = n$ class-E packets on arrival is $p_{m,n}$. Thus

$$\mathbb{E}[W_S 1_{\{\text{served}\}}] = \sum_{m,n} w(m, 0, n)p_{m,n}$$

and by the conditional probability formula it holds

$$\mathbb{E}[W_S|\text{served}] = \frac{\sum_{m,n} w(m, 0, n)p_{m,n}}{1-p_{\text{loss}}^{(S)}}. \tag{5}$$

The following proposition shows how to calculate the values $w(m, m', n)$. The proof is in Appendix B.

Proposition 3. *The conditional mean waiting times $w(m, m', n)$, $m + m' + n \le N - 1$ satisfy the recursive formulas*

$$\begin{aligned} w(m, m', n) = &\sum_{i+j\le N-m-m'-n-2} g_{ij}\left(c_{ij} + w(m-1, m'+i, n+j)\right) \\ &+ \sum_{\substack{i+j=N-m-m'-n-1 \\ k\le i+m'}} v_{ijk}\left(d_{ijk} + w(m-1, m'+i-k, n+j+k)\right) \\ w(0, m', n) = &\sum_{i+j\le N-m-m'-n-2} g_{ij}c_{ij} + \sum_{\substack{i+j=N-m-m'-n-1 \\ k\le i+m'}} v_{ijk}d_{ijk} \end{aligned}$$

where $g_{ij} = \binom{i+j}{i}\frac{\lambda_S^i\lambda_E^j\mu}{(\lambda_S+\lambda_E+\mu)^{i+j+1}}$, $c_{ij} = \frac{i+j+1}{\lambda_S+\lambda_E+\mu}$, $v_{ijk} = \binom{i+j}{i}\frac{\lambda_S^i\lambda_E^j}{(\lambda_S+\lambda_E+\mu)^{i+j}}\frac{(\alpha\lambda_E)^k\mu}{(\alpha\lambda_E+\mu)^{k+1}}$ *and* $d_{ijk} = \frac{i+j}{\lambda_S+\lambda_E+\mu} + \frac{k+1}{\alpha\lambda_E+\mu}$. *The above recursion finds the conditional mean waiting times $w(m, m', n)$ in $O(N^4)$ steps.*

In the special case when $\alpha = 1$, $w(m, m', n)$ does not depend on m', that is, the number of class-S packets that are in the queue after the tagged job do not affect the probability that the tagged job is served. Bear in mind that new class-E packets are not affected by them. In this case, the waiting time will only depend on the number of class-S jobs ahead of the tagged job, and the number of class-E jobs in the system. The waiting time could be calculated using Eq. (5) and Proposition 3. However, we can develop a more efficient algorithm by taking advantage of $w(m, n) = w(m, \cdot, n)$.

Proposition 4. *$w(m, n)$ satisfies:*

$$\begin{aligned} w(m, n) &= \sum_{k=0}^{N-n-m-1}(f_k + w(m-1, n+k)g_k), \\ w(0, n) &= \sum_{k=0}^{N-n-1} f_k, \quad w(idle) = 0 \end{aligned}$$

where $f_k = \frac{\lambda_E^k\mu}{(\lambda_E+\mu)^{k+2}}(k+1)$ *and* $g_k = \frac{\lambda_E^k\mu}{(\lambda_E+\mu)^{k+1}}$. *The recursive formulas are solved by index m (from 0 to N).*

Proof. We assume that the streaming packet that entered last will be pushed out first. First of all, let $\mathcal{P}$ be a Poison process with unitary rate and consider the formula

$$\begin{aligned} g_k &= \mathbb{E}\left[1_{\{\mathcal{P}(\lambda_E B)=k\}}\right] = \int_0^\infty \frac{(\lambda_E t)^k}{k!}e^{-\lambda_E t}f_B(t)dt \\ &= \int_0^\infty \frac{(\lambda_E t)^k}{k!}e^{-\lambda_E t}\mu e^{-\mu t}dt = \int_0^\infty \mu e^{-z}\frac{(\lambda_E\frac{z}{\lambda_E+\mu})^k}{k!}\frac{dz}{\lambda_E+\mu} \\ &= \frac{\lambda_E^k\mu}{(\lambda_E+\mu)^{k+1}k!}\Gamma(k+1) = \frac{\lambda_E^k\mu}{(\lambda_E+\mu)^{k+1}}, \end{aligned}$$

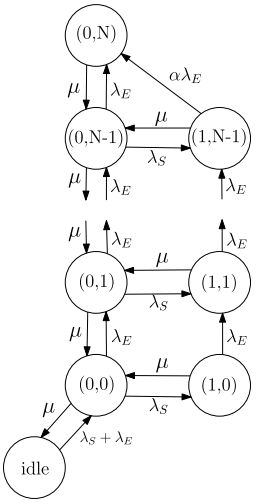


Fig. 2. State transition rate diagram for the light-traffic model. The diagram is a cut from the original STPQ where there will be at most two S packets in the system.

where $\Gamma(\cdot)$ is the Gamma function. The formula g_k expresses the probability that exactly k class-E packets arrive during a service time. Similarly, we obtain

$$\begin{aligned} f_k &= \mathbb{E}\left[B1_{\{\mathcal{P}(\lambda_E B)=k\}}\right] \\ &= \frac{\lambda_E^k \mu}{(\lambda_E+\mu)^{k+2}}(k+1). \end{aligned}$$

that gives the mean waiting time of the event defined by g_k, which takes $k+1$ steps with mean length $(\lambda_E+\mu)^{-1}$.

It is clear that $w(\text{idle}) = 0$, since a packet that arrives when the server is idle enters the service. Now, consider $m = 0$, when no streaming packet is waiting in the queue. In this case, the arriving class-S packets will receive service if at most $N-n-1$ elastic packets arrive before the one in service leaves. Thus, we can consider every described possibility and sum over them:

$$w(0,n) = \sum_{k=0}^{N-n-1} f_k.$$

For $m \geq 1$, considering $w(m',n)$ are known for every $m' < m$, note that being at state (m,n) and k elastic arrivals happen in a service time moves the system to the state $(m-1, n+k)$. With this, considering a streaming arrives at state (m,n) and knowing the probabilities that k elastic packets arrive (g_k), and the length of the service (f_k), we can derive

$$w(m,n) = \sum_{k=0}^{N-n-m-1} (f_k + w(m-1,n+k)g_k),$$

which can be computed since $w(m-1,n+k)$ are known from the previous steps. Solving for each m all the $w(m,n)$, we can do it again for $m+1$, until the system is completely solved. □

5. Light-traffic approximation

The light-traffic regime aims at approximating the steady-state performance when the traffic load is extremely low. More precisely, we will consider the regime in which $\lambda_S \to 0$, without making any assumption on the value of λ_E. As explained in the introduction, recent empirical measurements indicate that the arrival rate of streaming packets is much lower than that of elastic, and we can thus expect the light-traffic regime to be a good approximation in case the load due to streaming traffic is low. This approximation was pioneered in [23], see also [24]. In this regime, the number of jobs in the system will be very small. In particular, as shown in [24], the first-order light-traffic approximation is obtained by analyzing the original system in which at most one job may arrive. To explain this intuitively, note that if λ_S is very small, the probability of having two class-S arrivals is of order $o(\lambda_S^2)$, and thus negligible.

The goal of this section is to approximate the loss probabilities for both classes of packets, which from (2), (3), required finding $p_{0,N}$. The state transition rate diagram under the light traffic assumption is depicted in Fig. 2.

To solve the steady-state in the light-traffic regime, we will use known results from analytic perturbation theory applied to Markov Chains, see [25]. It is known that if the transition rate matrix and the equilibrium probabilities are

of the form $Q(\epsilon) = Q_0 + \epsilon Q_1 + \epsilon^2 Q_2 + \cdots$ and $\pi(\epsilon) = \pi_0 + \epsilon\pi_1 + \epsilon^2\pi_2 + \cdots$, then the steady-state probabilities can be obtained by solving the system of fundamental equations

$$\begin{aligned} \pi_0 Q_0 &= 0 \\ \pi_1 Q_0 + \pi_0 Q_1 &= 0 \\ &\vdots \\ \pi_k Q_0 + \pi_{k-1} Q_1 + \cdots + \pi_1 Q_{k-1} + \pi_0 Q_k &= 0 \\ &\vdots \end{aligned}$$

together with the system normalization conditions

$$\pi_0 \mathbf{1} = 1$$
$$\pi_k \mathbf{1} = 0, \ k \geq 1$$

We will apply this characterization to our queueing model. For our situation, with the state space ordered {idle, (0, 0), (0, 1), ..., (0, $N-1$), (0, N), (1, 0), (1, 1), ..., (1, $N-2$), (1, $N-1$)} the transition rate matrix Q^{LT} can be written as

$$Q^{LT}(\lambda_S) = Q_0 + \lambda_S Q_1,$$

where

$$Q_0 = \left(\begin{array}{cccc|cccc} & Q_E^{M/M/1/N+1} & & & & 0 & & \\ \hline 0 & \mu & & & -(\lambda_E+\mu) & \lambda_E & & \\ \vdots & \ddots & & & & \ddots & \ddots & \\ 0 & & \mu & \alpha\lambda_E & & & & -(\alpha\lambda_E+\mu) \end{array}\right)$$

and

$$Q_1 = \left(\begin{array}{cccc|ccc} -1 & 1 & & & 0 & \dots & 0 \\ & -1 & & & 1 & & \\ & & \ddots & & & \ddots & \\ & & & -1 & & & 1 \\ & & & 0 & & & 0 \\ \hline & & 0 & & & 0 & \end{array}\right),$$

where $Q_E^{M/M/1/N+1}$ denotes the transition rates of an $M/M/1/N+1$ queue with only class-E packets.

We are interested in approximating the only unknown term in the loss probability equations (2), (3), which is $p_{0,N}$. In the light-traffic regime, p_N will be composed by $p_{0,N} + p_{1,N-1}$. From our knowledge of p_N from the $M/M/1/N+1$ system and the light-traffic approximation for $p_{1,N-1}$ developed in the following Lemma 1, we can obtain an approximation for $p_{0,N}$. The proof is deferred to Appendix C.

Lemma 1. *The light-traffic approximation of the STPQ queue for $p_{0,N}$ is*

$$p_{0,N}^{LT} = p_N - p_{1,N-1}^{LT} \tag{6}$$

where

$$p_{1,N-1}^{LT} = \frac{\lambda_S}{\alpha\lambda_E + \mu} \frac{1-\rho_E}{1-\rho_E^{N+2}} \left(\rho_E^{N-1} - \left(\frac{\lambda_E}{\lambda_E + \mu} \right)^{N-1} + \rho_E^N \right)$$

The accuracy of the light-traffic approximation for different system configurations is analyzed in Section 6.3.

6. Numerical results

In this section, we will illustrate the performance of the space–time priority queue by means of numerical examples.

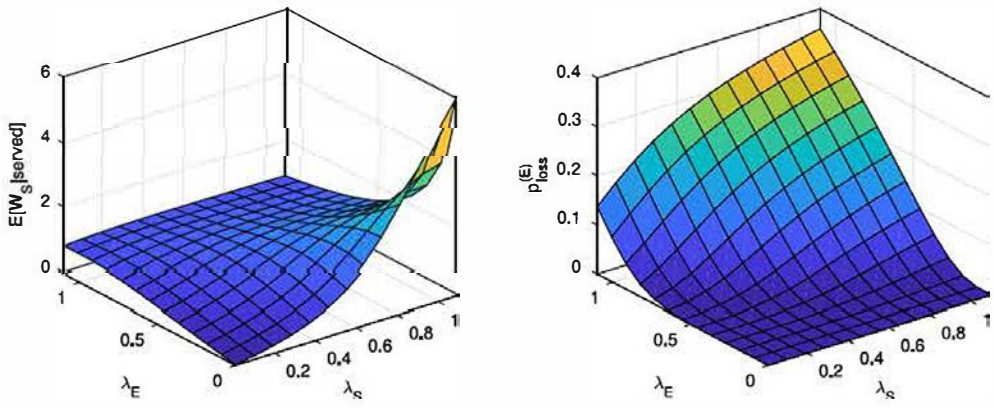


Fig. 3. Mean waiting time *(left)* and loss probability *(right)* in the non-randomized push-out STPQ ($\alpha = 1$) with $N = 10$, $\mu = 1$ and different traffic configurations.

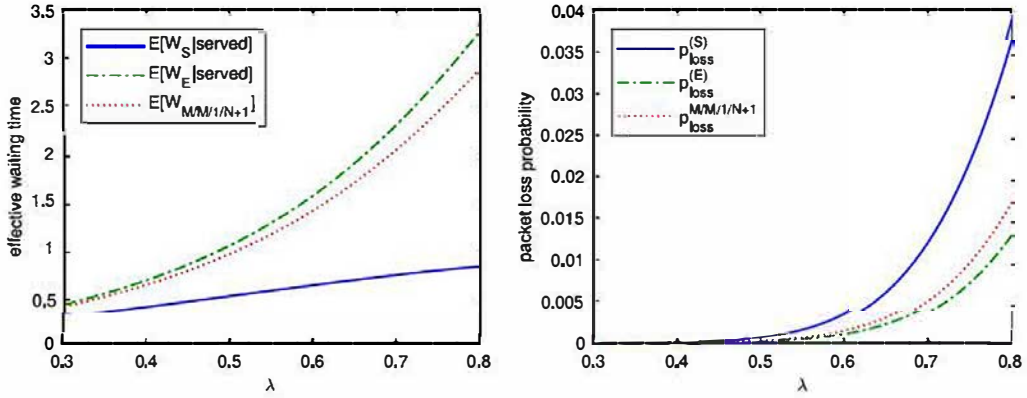


Fig. 4. Mean waiting time *(left)* and loss probability *(right)* with $N = 10$ and $\alpha = 1$. Proportion of traffic satisfies $\lambda_S/\lambda_E = 0.2$ and total arrival rate is $\lambda = \lambda_S + \lambda_E$.

6.1. Case $\alpha = 1$

In this subsection we study the loss probability and the waiting time in the case $\alpha = 1$.

As we can see in Fig. 3 *(left)*, the mean waiting time for streaming packets remains low over almost every configuration of the system. When elastic traffic is close to 0, the waiting time is increasing on λ_S, since we would be just dealing with an $M/M/1/N+1$ queue for class-S. We also see that when the arrival rate of both classes is high, the waiting time for the streaming packets that are not pushed-out does not increase. Even if class-E traffic is high, we observe that the waiting time of class-S does not increase significantly. This is due to the fact that we measure the waiting time conditioned on the packet being served. Thus, we observe that the only scenario in which class-S packets have a large waiting time is when λ_S is large.

In Fig. 3 *(right)* we plot the loss probabilities for different configurations. We observe that for small to moderate values of λ_E, the loss probability of class-E packets is fairly insensitive to the value of λ_S.

As in [3], we now assume that the ratio of streaming traffic over elastic traffic is fixed with value $\lambda_S/\lambda_E = 0.2$. Numerical results of this scenario for different traffic intensities can be observed in Fig. 4. In Fig. 4 *(left)*, it can be seen that, while traffic becomes high, waiting time for streaming class is much lower than the one in the system without priorities, that is, the $M/M/1/N+1$ system. Moreover, the waiting time for elastic packets is not much bigger than in the non priority queue. Then, in Fig. 4 *(right)*, the loss probabilities are represented for the same parameter settings. There, it is seen that for high arrival rates, the performance of elastic traffic regarding the loss probability improves with respect to non priority system, and that the contrary holds for the streaming packets.

6.2. The case $0 \leq \alpha \leq 1$

By varying the traffic rates of streaming and elastic class packets, λ_S and λ_E, we consider different scenarios to analyze the loss probabilities of the randomized push-out mechanism by means of the control parameter α.

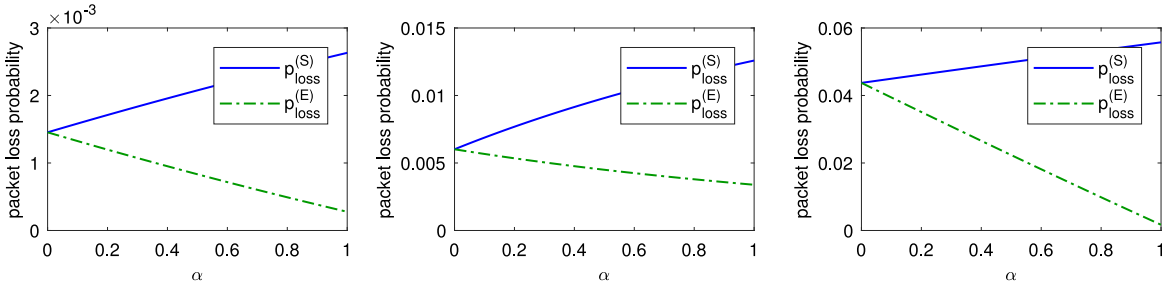


Fig. 5. Loss probabilities for the STPQ with parameters $N = 10$ and $\mu = 1$ with respect to the push-out probability α for different traffic configurations. $\lambda_S = \lambda_E = 0.3$ *(left)*, $\lambda_S = 0.2,\ \lambda_E = 0.5$ *(center)* and $\lambda_S = 0.7,\ \lambda_E = 0.2$ *(right)*.

Scenario 1: similar traffic intensities

We suppose in this first scenario that we are dealing with similar traffic intensities for both streaming and elastic packets. We let the parameters be $N = 10,\ \lambda_S = 0.3,\ \lambda_E = 0.3,\ \mu = 1$.

For a situation in which the total amount of traffic intensity is low, see Fig. 5 *(left)*, loss probabilities are of order 10^{-3} for both classes. As we can see, while α grows from zero to one the increase of class-S loss probabilities and the decrease of class-E loss probabilities are roughly of the same order.

We should note that the loss probabilities are the same for both types of traffic when $\alpha = 0$. In that situation there is no push-out, so every packet that arrives when the buffer is full will be refused. Thus, the loss probability does not depend on the incoming packet class, only on the resulting traffic rate (and the size of the buffer). Moreover, this loss probability is the probability that $M/M/1/N{+}1$ is full, given by p_N.

We conclude that in this scenario where streaming and elastic traffic is similar, the parameter α can be adapted in order to reduce the probability for one class whereas for the other class is increased. This allows the controller to reduce the loss probability for one class without sacrificing the other class traffic.

Scenario 2: higher traffic intensity of streaming packets

In this scenario we analyze the situation in which $\lambda_S > \lambda_E$. We consider for this scenario $N = 10,\ \lambda_S = 0.7,\ \lambda_E = 0.2,\ \mu = 1$. In Fig. 5 *(right)* we can see that the behavior in not the same as in Scenario 1. The relevant property of this scenario is that increasing the push-out probability improves a lot the loss probability of class-E packets without having a great impact into the class-S packets.

Scenario 3: higher traffic intensity of elastic packets

In this scenario we take the parameters $N = 10$, $\lambda_S = 0.2$, $\lambda_E = 0.5$, $\mu = 1$. In this case, enabling the push-out mechanism with a high probability α reduces the probability of losing class-E packets but the price is high for class-S packets. We can see again in Fig. 5 *(center)* how, with the growth of α, the loss probability for streaming packets increases very fast in comparison to the loss probability reduction for elastic packets.

In addition to the loss probabilities, we also consider the waiting times of class-S and class-E packets in the case of $0 \leq \alpha \leq 1$. In Fig. 6 we consider the same traffic configuration as in Fig. 4, but with $\alpha = 0.3$. Comparing the results with respect to the non-priority case, that is, the $M/M/1/N{+}1$ case, we see that *(left)* the mean waiting time of class-S packets improves considerably whereas *(right)* the degradation of the loss probability of class-S packets is less significant than in the case $\alpha = 1$. In Fig. 7 we plot the waiting time of class E and S packets. We observe that the waiting time is largely insensitive to the value of α. Thus, it can be concluded that the push-out probability parameter α gives flexibility to optimize the waiting time of streaming packets, while keeping control of the blocking probability of class-E packets.

The results of Fig. 7 also suggest that instead of making use of the algorithm proposed in Proposition 3, the waiting time could be efficiently approximated by a linear interpolation between the values for $\alpha = 0$ and $\alpha = 1$. The case $\alpha = 0$, that is, the finite queue with time-priority, can be computed using the Little's law for class-S packets because in this particular case, any of those packets that enter the queue will receive service. For $\alpha = 1$, the algorithm proposed in Proposition 4 may be used.

6.3. Accuracy of light-traffic approximation

We analyze now the accuracy of our light-traffic approach. In Fig. 8 *(left)* we consider a scenario where the proportion of class-S packets over class-E ones is 0.15. We observe that the error for class-S packets is higher than for class-E packets, but the approximation is considerably accurate in both cases. In Fig. 8 *(right)* we consider a scenario with the proportion 0.3. In this case, we can observe that the accuracy of the light-traffic regime is slightly lower than before, as we are moving

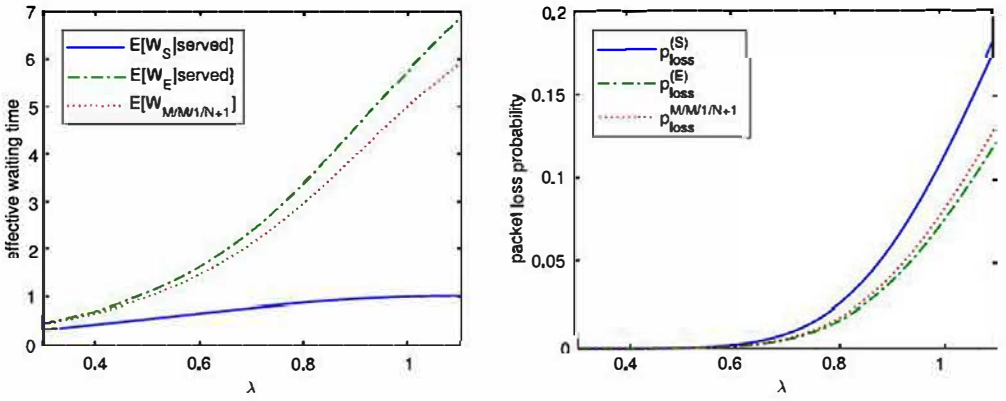


Fig. 6. Mean waiting time *(left)* and loss probability *(right)* of the STPQ with $N = 10$, $\mu = 1$ and $\alpha = 0.3$. The arrival rate is $\lambda = \lambda_S + \lambda_E$ with $\lambda_S/\lambda_E = 0.2$.

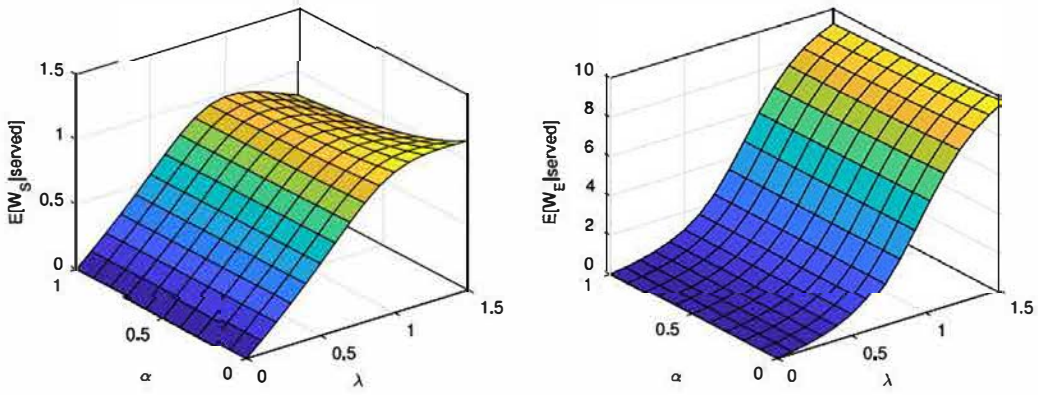


Fig. 7. Mean waiting time for class-S *(left)* and class-E *(right)* packets with $N = 10$, $\mu = 1$ and a proportion of traffic $\lambda_S/\lambda_E = 0.25$, with total arrival rate $\lambda = \lambda_S + \lambda_E$.

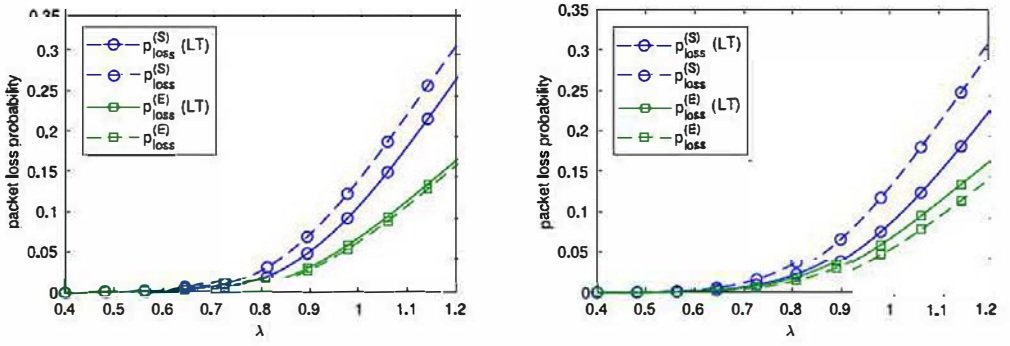


Fig. 8. Light-traffic approximation performance over loss probabilities for both class-S and class-E packets on a system with parameters $N = 10$, $\mu = 1$ and $\alpha = 0.8$. The arrival rate is $\lambda = \lambda_S + \lambda_E$ with proportions satisfying *(left)* $\lambda_S/\lambda_E = 0.15$ and *(right)* $\lambda_S/\lambda_E = 0.3$.

to a configuration in which class-S traffic is not as light as it was in the previous configuration. We note that this traffic ratio is considerably more balanced than the one observed in real traces.

7. Simulation results

In this section, results of simulations are presented to evaluate scenarios where previously stated assumptions do not hold. We program a simulator in order to assess the performance of STPQ with non-Markovian assumptions for two different scenarios, one considering deterministic packet sizes and the other from a real trace.

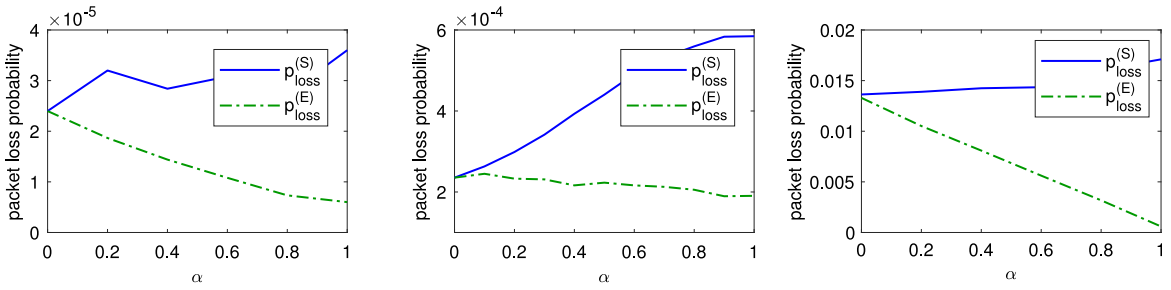


Fig. 9. Loss probabilities for the STPQ with parameters $N = 10$ and deterministic service times (size 1) with respect to the push-out probability α for different traffic configurations. $\lambda_S = \lambda_E = 0.3$ *(left)*, $\lambda_S = 0.2$, $\lambda_E = 0.5$ *(center)* and $\lambda_S = 0.7$, $\lambda_E = 0.2$ *(right)*.

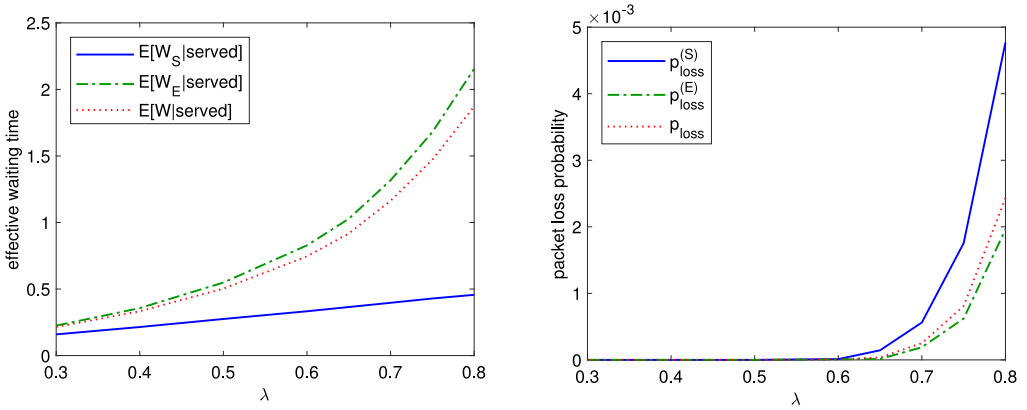


Fig. 10. Mean waiting time *(left)* and loss probability *(right)* with $N = 10$, $\alpha = 1$ and deterministic service times (size 1). Proportion of traffic satisfies $\lambda_S/\lambda_E = 0.2$ and total arrival rate is $\lambda = \lambda_S + \lambda_E$.

7.1. Deterministic packet sizes

Throughout this subsection, it is considered that packets arrive under the Markovian assumption and that packet sizes are deterministic of size 1.

Fig. 9 shows the loss probability for the same parameters as the ones considered in Fig. 5. Comparing the two sets of figures, it can be seen that the curves show qualitatively a similar shape, with loss probabilities increasing and decreasing in similar proportions for the Markovian and non-Markovian settings, and that in the deterministic case it is less likely to loss packets, which is expectable due to the lower variability of the deterministic service time distribution.

Similar conclusion can be drawn when comparing the performance as a function of the arrival rate, see Figs. 10 and 11 for deterministic packet sizes and Figs. 4 and 6 for exponentially distributed packets. While loss probabilities are lower as a function of the arrival rate, the effective waiting time does not show the same pattern. For low arrival rates, the waiting time is smaller with deterministic packets sizes, but as the arrival rate increases, for example greater than 1 in Fig. 11, class-E packets have to wait more in the case packets are deterministic.

Having analyzed the different scenarios for the STPQ with deterministic service times, we conclude that the formulas derived for the Markovian setting provide a good baseline to estimate the qualitative behavior of the STPQ.

7.2. Real trace from WIDE

In this subsection we consider a trace from WIDE with measurements from 2019/01/02, see [4]. The data set we consider is formed by 5 million packets, from which 95% are TCP (type E) and the remaining 5% UDP (type S). In Fig. 12 we visualize the distribution of the *(left)* packet inter-arrival time and the *(right)* packet size. We observe that the inter arrival time for TCP packets is much smaller, which is expected given that 95% of the packets TCP. Another interesting observation is that TCP packets are significantly larger than UDP ones. For the simulation, we consider a server with speed 1 Gigabyte per second as specified in the source of the trace, and the buffer size was chosen to be $N = 4$ so that the impact of the push-out probability parameter α can be observed.

Fig. 13 shows the performance of the STPQ for the defined trace. In *(left)* we see that the waiting time of class-S or UDP packets is considerably smaller than that of class-E or TCP packets. We also note that the choice of α does not have

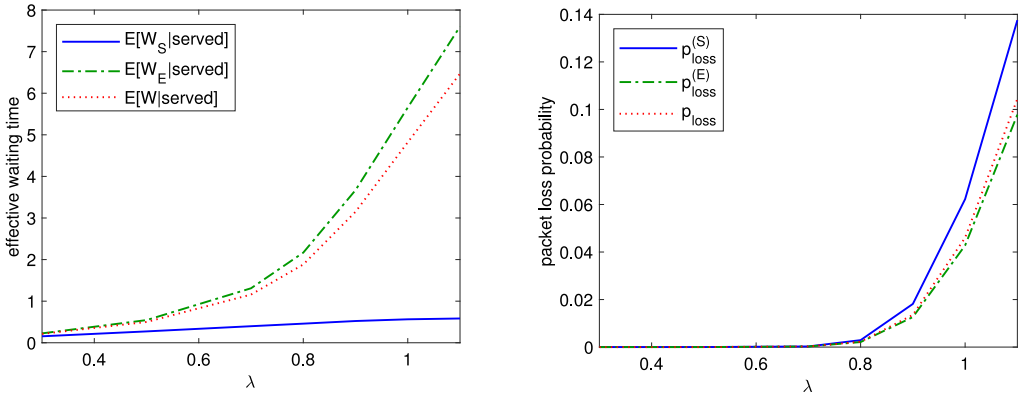


Fig. 11. Mean waiting time *(left)* and loss probability *(right)* of the STPQ with $N = 10$, deterministic service times (size 1) and $\alpha = 0.3$. The arrival rate is $\lambda = \lambda_S + \lambda_E$ with $\lambda_S/\lambda_E = 0.2$.

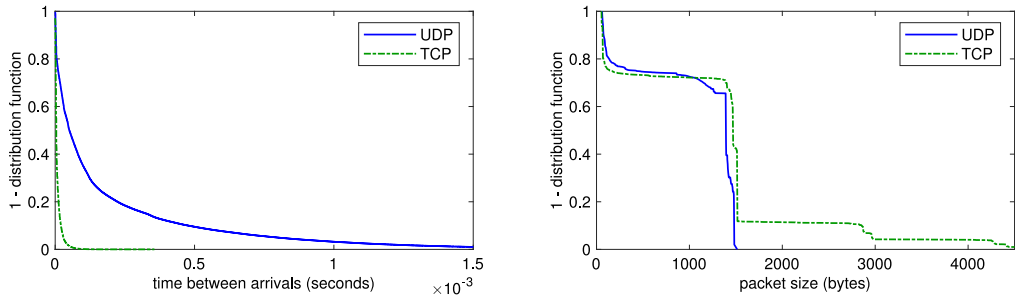


Fig. 12. Complementaries of the cumulative distribution functions of seconds between packet arrivals *(left)* and size of packets in seconds *(right)* of the real trace of 5 million packets considered to simulate the STPQ.

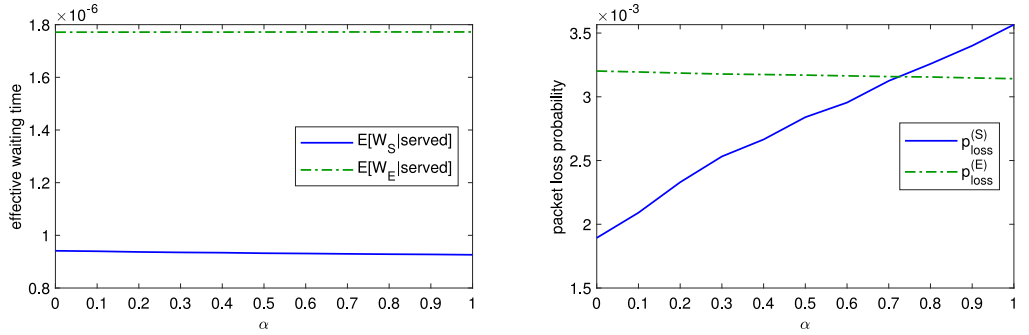


Fig. 13. Mean waiting time *(left)* and loss probability *(right)* of the STPQ with $N = 4$ for a real trace of 5 million packets processed at a rate of 1 Gigabyte per second.

much of an impact. On the *(right)* we see that the packet loss probability clearly increases on α for UDP traffic, whereas it decreases at a lower rate for TCP packets.

These observations regarding the impact of α coincide qualitatively, under similar choice of other parameters, with the predictions of our mathematical model for the Markovian setting.

8. Conclusions

In this paper we have introduced and studied the space–time priority queue (STPQ), where one class has time priority, whereas the other one has space priority. The main motivation of STPQ is to model the performance of scheduling algorithms that can provide service priority to streaming type of packets (say UDP traffic in the Internet) and space priority to elastic type of packets (say TCP). In the literature such scheduling algorithms have been previously proposed, see [5,6], but to the best of our knowledge, our work is the first attempt to propose a queueing model to analyze it. In our main contribution we have derived algorithms to calculate efficiently the loss probabilities and the mean waiting time. The

numerical examples show that STPQ permits, comparing to a queue without priorities, to provide a lower delay to class-S packets without a significant degradation of the performance of class-E packets.

Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.peva.2019.04.003.

Appendix A. Proof of Proposition 1

We recall from Section 3 that the total probability along diagonals are known, as well as p_{idle} and $p_{0,0}$. We first focus on the balance equations for states $(1, 0)$ to $(N, 0)$. We have,

$$\begin{aligned} p_{i,0}(\lambda_E + \lambda_S + \mu) &= p_{i+1,0}\mu + p_{i-1,0}\lambda_S, \quad i = 1, \ldots, N-1, \\ p_{N,0}(\alpha\lambda_E + \mu) &= p_{0,N-1}\lambda_S . \end{aligned}$$

As $p_{0,0}$ is already known, we have N equations and N unknowns. We will solve the system in $O(N)$ (i.e. linear order complexity). We put $p_{0,N}$ in the following form

$$p_{0,N} = \beta p_{0,0}$$

Then, from the balance equations it follows that

$$\begin{aligned} p_{N-1,0} &= p_{N,0}\frac{\alpha\lambda_E + \mu}{\lambda_S} = \frac{\alpha\lambda_E + \mu}{\lambda_S}\beta p_{0,0}, \\ p_{N-2,0} &= p_{N-1,0}\frac{\lambda_E + \lambda_S + \mu}{\lambda_S} - p_{N,0}\frac{\mu}{\lambda_S} \\ &= \left(\frac{\alpha\lambda_E + \mu}{\lambda_S}\frac{\lambda_E + \lambda_S + \mu}{\lambda_S} - \frac{\mu}{\lambda_S}\right)\beta p_{0,0}, \\ &\vdots \end{aligned}$$

In other words, if we define b_i for $i = N$ to $i = 0$ with the following recursion,

$$\begin{aligned} b_N &= 1, \quad b_{N-1} = \frac{\alpha\lambda_E + \mu}{\lambda_S} \\ b_i &= b_{i+1}\frac{\lambda_E + \lambda_S + \mu}{\lambda_S} - b_{i+2}\frac{\mu}{\lambda_S} . \end{aligned}$$

then

$$p_{i,0} = b_i\beta p_{0,0}.$$

From the equation for the case of $p_{0,0}$ we can derive the value of β.

$$\beta = \frac{1}{b_0}.$$

So, if we let

$$\beta_0 = \frac{p_{0,0}}{b_0} = \frac{(1-\rho)\rho}{b_0(1-\rho^{N+2})}$$

it is verified that the values $p_{i,0} = b_i\beta_0$ solve the system of equations.

We can now follow a similar procedure for the following rows in the transition diagram. For the j-th row, we can assume that we know all probabilities $p_{i,k}$ for $k < j$. Moreover, we can determine $p_{0,j}$ by summing the jth diagonal,

$$p_{0,j} = \frac{(1-\rho)\rho^{j+1}}{1-\rho^{N+2}} - \sum_{i=1}^{j} p_{i,j-i} .$$

The $N - j$ equations we consider are,

$$\begin{aligned} p_{i,j}(\lambda_E + \lambda_S + \mu) &= p_{i+1,j}\mu + p_{i-1,j}\lambda_S + p_{i,j-1}\lambda_E \\ p_{N-j,j}(\alpha\lambda_E + \mu) &= p_{N-j-1,j}\lambda_S + p_{N-j,j-1}\lambda_E + p_{N-j+1,j-1}\alpha\lambda_E \end{aligned}$$

For these equations, we conclude that it will not be a proportional rule for the states of the row since there is a dependency over the probabilities solved in other rows. Thus, we will have now

$$p_{N-j,j} = \varphi + \beta p_{0,j}.$$

To solve these equations in $O(N-j)$ we go downwards as we did for the first row. We have to calculate the values $a_i^{(j)}$ and $b_i^{(j)}$ by the recursion,

$$a_i^{(j)} = a_{i+1}^{(j)} \frac{\lambda_E + \lambda_S + \mu}{\lambda_S} - a_{i+2}^{(j)} \frac{\mu}{\lambda_S} - p_{i+1,j-1} \frac{\lambda_E}{\lambda_S},$$
$$b_i^{(j)} = b_{i+1}^{(j)} \frac{\lambda_E + \lambda_S + \mu}{\lambda_S} - b_{i+2}^{(j)} \frac{\mu}{\lambda_S},$$

with initial values,

$$a_{N-j}^{(j)} = 0, \qquad a_{N-j-1}^{(j)} = -(p_{N-j,j-1} + \alpha p_{N-j+1,j-1}) \frac{\lambda_E}{\lambda_S},$$
$$b_{N-j}^{(j)} = 1, \qquad b_{N-j-1}^{(j)} = \frac{\alpha\lambda_E + \mu}{\lambda_S}.$$

Noting that the solution to the recursion $b_i^{(j)}$ is $b_i^{(j)} = b_{i+j}$ and then introducing the factor β_j,

$$\beta_j = \frac{p_{0,j} - a_0^{(j)}}{b_j},$$

one verifies that the set of equations is solved by the probabilities $p_{i,j} = a_i^{(j)} + b_{i+j}\beta_j$.

Appendix B. Proof of Proposition 3

Let us consider the situation where the tagged packet is at state (m, m', n), for $m+m'+n \leq N-1$. The value of interest is

$$w(m, m', n) = \mathbb{E}\left[W_S 1_{\{\text{served}\}} \mid m \text{ class-}S \text{ before, } m' \text{ class-}S \text{ after, } n \text{ class-}E\right].$$

We will write a recursive equation $w(m, m', n)$ by keeping track of the arrivals during the service time.

We first consider the case when the queue is not full when the service time finishes, that is, $m+1+m'+n+i+j \leq N-1$, where i is number of arrivals of class-S, and j arrivals of class-E. Let us denote by g_{ij} the probability that during the service time there are i arrivals of class-S and j arrivals of class-E, i.e.,

$$g_{ij} = \mathbb{E}\left[1_{\{\mathcal{P}(\lambda_S B)=i, \mathcal{P}(\lambda_E B)=j\}}\right],$$

where $\mathcal{P}$ denotes a Poisson process with unitary rate. Similarly we also define

$$f_{ij} = \mathbb{E}\left[B 1_{\{\mathcal{P}(\lambda_S B)=i, \mathcal{P}(\lambda_E B)=j\}}\right].$$

Consider now that at the time of the next service, the queue is full, that is, there has arrived i type S and j type E arrivals with $m+1+m'+n+i+j = N$. Let S_{i+j} be the random variable denoting the time of the $i+j$-th arrival of a Poisson process with rate $\lambda_S + \lambda_E$, and let v_{ijk} the probability that during a service time i class-S and j class-E packets arrive, and that additional k class-E packets push-out a class-S packet, i.e.,

$$v_{ijk} = \mathbb{E}\left[1_{\{\mathcal{P}(\lambda_S S_{i+j})=i, \mathcal{P}(\lambda_E S_{i+j})=j, S_{i+j}<B, \mathcal{P}(\alpha\lambda_E(B-S_{i+j}))=k\}}\right].$$

Similarly, we define:

$$u_{ijk} = \mathbb{E}\left[B 1_{\{\mathcal{P}(\lambda_S S_{i+j})=i, \mathcal{P}(\lambda_E S_{i+j})=j, S_{i+j}<B, \mathcal{P}(\alpha\lambda_E(B-S_{i+j}))=k\}}\right].$$

We can now write the recursion:

$$w(m, m', n) = \sum_{i+j \leq N-m-m'-n-2} \left(f_{ij} + g_{ij} w(m-1, m'+i, n+j)\right) + \sum_{\substack{i+j=N-m-m'-n-1 \\ k \leq i+m'}} \left(u_{ijk} + v_{ijk} w(m-1, m'+i-k, n+j+k)\right) \tag{B.1}$$

together with $w(m, m', n) = 0$ for $m < 0$. This provides a set of recursive formulas over m, from zero to $N-1$, to compute the values of $w(m, m', n)$.

We now calculate the expressions for f_{ij}, g_{ij}, v_{ijk} and u_{ijk}.

$$f_{ij} = \mathbb{E}\left[B 1_{\{\mathcal{P}(\lambda_S B)=i, \mathcal{P}(\lambda_E B)=j\}}\right] = \int_0^\infty t f_B(t) \mathbb{P}(\mathcal{P}(\lambda_S t) = i, \mathcal{P}(\lambda_E t) = j) dt$$
$$= \int_0^\infty t\mu e^{-\mu t} \mathbb{P}(\mathcal{P}(\lambda_S t) = i) \mathbb{P}(\mathcal{P}(\lambda_E t) = j) dt = \int_0^\infty \frac{\lambda_S^i \lambda_E^j \mu}{i!j!} t^{i+j+1} e^{-(\lambda_S+\lambda_E+\mu)t} dt$$

$$= \frac{\lambda_S^i \lambda_E^j \mu}{i!j!(\lambda_S+\lambda_E+\mu)^{i+j+1}} \int_0^\infty z^{i+j+1} e^{-z} dz = \frac{\lambda_S^i \lambda_E^j \mu}{i!j!(\lambda_S+\lambda_E+\mu)^{i+j+1}} \Gamma(i+j+2)$$
$$= \binom{i+j}{i} \frac{\lambda_S^i \lambda_E^j \mu (i+j+1)}{(\lambda_S+\lambda_E+\mu)^{i+j+2}}, \ i \geq 0, j \geq 0,$$

where $\Gamma(\cdot)$ is the Gamma function. Similarly,

$$g_{ij} = \mathbb{E}\left[1_{\{\mathcal{P}(\lambda_S B)=i, \mathcal{P}(\lambda_E B)=j\}}\right] = \int_0^\infty f_B(t) \mathbb{P}(\mathcal{P}(\lambda_S t)=i, \mathcal{P}(\lambda_E t)=j) dt$$
$$= \frac{\lambda_S^i \lambda_E^j \mu}{i!j!} \int_0^\infty t^{i+j} e^{-(\lambda_S+\lambda_E+\mu)t} dt = \binom{i+j}{i} \frac{\lambda_S^i \lambda_E^j \mu}{(\lambda_S+\lambda_E+\mu)^{i+j+1}}, \ i \geq 0, j \geq 0.$$

We now proceed similarly to calculate u_{ijk} and v_{ijk}. We have

$$u_{ijk} = \mathbb{E}\left[B 1_{\{\mathcal{P}(\lambda_S S_{i+j})=i, \mathcal{P}(\lambda_E S_{i+j})=j, S_{i+j}<B, \mathcal{P}(\alpha\lambda_E(B-S_{i+j}))=k\}}\right]$$
$$= \int_0^\infty t f_B(t) \int_0^t f_{S_{i+j}|\mathcal{P}((\lambda_S+\lambda_E)s)=i+j}(s) \mathbb{P}(\mathcal{P}(\lambda_S s)=i) \mathbb{P}(\mathcal{P}(\lambda_E s)=j) \mathbb{P}(\mathcal{P}(\alpha\lambda_E(t-s))=k) ds dt$$
$$= \int_0^\infty t\mu e^{-\mu t} \int_0^t \frac{i+j}{s} \frac{\lambda_S^i \lambda_E^j s^{i+j} e^{-(\lambda_S+\lambda_E)s}}{i!j!} \frac{(\alpha\lambda_E)^k (t-s)^k e^{-\alpha\lambda_E(t-s)}}{k!} ds dt$$
$$= \frac{\lambda_S^i \lambda_E^j (\alpha\lambda_E)^k \mu (i+j)}{i!j!k!} \int_0^\infty s^{i+j-1} e^{-(\lambda_S+\lambda_E)s} \int_s^\infty t e^{-\mu t} (t-s)^k e^{-\alpha\lambda_E(t-s)} dt ds$$
$$= \frac{\lambda_S^i \lambda_E^j (\alpha\lambda_E)^k \mu (i+j)}{i!j!k!} \int_0^\infty s^{i+j-1} e^{-(\lambda_S+\lambda_E+\mu)s} \int_0^\infty (s+z) z^k e^{-(\alpha\lambda_E+\mu)z} dz ds$$
$$= \frac{\lambda_S^i \lambda_E^j (\alpha\lambda_E)^k \mu (i+j)}{i!j!k!} \int_0^\infty s^{i+j-1} e^{-(\lambda_S+\lambda_E+\mu)s} \left(\frac{s\Gamma(k+1)}{(\alpha\lambda_E+\mu)^{k+1}} + \frac{\Gamma(k+2)}{(\alpha\lambda_E+\mu)^{k+2}} \right) ds$$
$$= \frac{\lambda_S^i \lambda_E^j (\alpha\lambda_E)^k \mu (i+j)}{i!j!(\alpha\lambda_E+\mu)^{k+1}} \int_0^\infty s^{i+j-1} e^{-(\lambda_S+\lambda_E+\mu)s} \left(s + \frac{k+1}{\alpha\lambda_E+\mu} \right) ds$$
$$= \frac{\lambda_S^i \lambda_E^j (\alpha\lambda_E)^k \mu (i+j)}{i!j!(\alpha\lambda_E+\mu)^{k+1}} \left(\frac{\Gamma(i+j+1)}{(\lambda_S+\lambda_E+\mu)^{i+j+1}} + \frac{k+1}{\alpha\lambda_E+\mu} \frac{\Gamma(i+j)}{(\lambda_S+\lambda_E+\mu)^{i+j}} \right)$$
$$= \binom{i+j}{i} \frac{\lambda_S^i \lambda_E^j}{(\lambda_S+\lambda_E+\mu)^{i+j}} \frac{(\alpha\lambda_E)^k \mu}{(\alpha\lambda_E+\mu)^{k+1}} \left(\frac{i+j}{\lambda_S+\lambda_E+\mu} + \frac{k+1}{\alpha\lambda_E+\mu} \right), \ i \geq 0, j \geq 0, k \geq 0$$

and

$$v_{ijk} = \mathbb{E}\left[1_{\{\mathcal{P}(\lambda_S S_{i+j})=i, \mathcal{P}(\lambda_E S_{i+j})=j, S_{i+j}<B, \mathcal{P}(\alpha\lambda_E(B-S_{i+j}))=k\}}\right]$$
$$= \int_0^\infty f_B(t) \int_0^t f_{S_{i+j}|P((\lambda_S+\lambda_E)s)=i+j}(s) \mathbb{P}(\mathcal{P}(\lambda_S s)=i) \mathbb{P}(\mathcal{P}(\lambda_E s)=j) \mathbb{P}(\mathcal{P}(\alpha\lambda_E(t-s))=k) ds dt$$
$$= \frac{\lambda_S^i \lambda_E^j (\alpha\lambda_E)^k \mu (i+j)}{i!j!k!} \int_0^\infty s^{i+j-1} e^{-(\lambda_S+\lambda_E)s} \int_s^\infty e^{-\mu t} (t-s)^k e^{-\alpha\lambda_E(t-s)} dt ds$$
$$= \binom{i+j}{i} \frac{\lambda_S^i \lambda_E^j}{(\lambda_S+\lambda_E+\mu)^{i+j}} \frac{(\alpha\lambda_E)^k \mu}{(\alpha\lambda_E+\mu)^{k+1}}, \ i \geq 0, j \geq 0, k \geq 0.$$

We can now finish the proof by rewriting Equation (B.1) as

$$w(m, m', n) = \sum_{i+j \leq N-m-m'-n-2} g_{ij} \left(c_{ij} + w(m-1, m'+i, n+j) \right) \tag{B.2}$$
$$+ \sum_{\substack{i+j=N-m-m'-n-1 \\ k \leq i+m'}} v_{ijk} \left(d_{ijk} + w(m-1, m'+i-k, n+j+k) \right) \tag{B.3}$$

where $c_{ij} = \frac{i+j+1}{\lambda_S+\lambda_E+\mu}$ and $d_{ijk} = \frac{i+j}{\lambda_S+\lambda_E+\mu} + \frac{k+1}{\alpha\lambda_E+\mu}$.

Finally, it is easy to check that (B.3) expresses $w(m, m', n)$ in terms of $w(m-1, \cdot, \cdot)$ so that (B.3) can be used to recursively calculate all w's by iterating over m. There are $O(N^3)$ unknown w's, while it takes $O(N)$ operations per w, yielding a numerical complexity of $O(N^4)$.

Appendix C. Proof of Lemma 1

We denote p^{LT} the row which contains the equilibrium probabilities for our light-traffic regime model, that is

$$p^{LT} = [p_{\text{idle}}, p_{0,0}, \ldots, p_{0,N}, p_{1,0}, \ldots, p_{1,N-1}]^{LT}$$

and we suppose

$$p^{LT} = p^{(0)} + \lambda_S p^{(1)} + \cdots$$

Then, invoking the analytic perturbation results of [25] we get:

$$p^{(0)} Q_0 = 0 \tag{C.1}$$

$$p^{(0)} Q_1 + p^{(1)} Q_0 = 0 \tag{C.2}$$

Solving (C.1) we get for $j = 0, 1, \ldots, N$,

$$p^{(0)}_{0,j} = \frac{1-\rho_E}{1-\rho_E^{N+2}} \rho_E^{j+1},$$

$$p^{(0)}_{1,j} = 0.$$

The solution could have been known before solving the system. The interpretation is that $p^{(0)}$ are the probabilities when the arrival rate of the streaming packets is zero and then the solution is same as the $M/M/1/N{+}1$ queue for the elastic class packets.

Now, for the next part we have

$$p^{(0)} Q_1 = [\ldots\ldots \vdots, p^{(0)}_{0,0}, p^{(0)}_{0,1}, \ldots, p^{(0)}_{0,N-1}],$$

$$p^{(1)} Q_0$$

$$= [\ldots\ldots \vdots, -(\lambda_E+\mu)p^{(1)}_{1,0}, \lambda_E p^{(1)}_{1,0} - (\lambda_E+\mu)p^{(1)}_{1,1}, \ldots, \lambda_E p^{(1)}_{1,N-3} - (\lambda_E+\mu)p^{(1)}_{1,N-2}, \lambda_E p^{(1)}_{1,N-2} - (\alpha\lambda_E+\mu)p^{(1)}_{1,N-1}].$$

Note that we are only taking into account the second part of the vectors, that is, the slots of the vector related to the states in which one streaming is waiting for service: $(1,0), (1,1), \ldots, (1, N-1)$. Substituting in (C.2) we get:

$$p^{(0)}_{0,0} - (\lambda_E+\mu)p^{(1)}_{1,0} = 0,$$

$$p^{(0)}_{0,j} + \lambda_E p^{(1)}_{1,j-1} - (\lambda_E+\mu)p^{(1)}_{1,j} = 0, \quad j = 1, \ldots, N-2,$$

$$p^{(0)}_{0,N-1} + \lambda_E p^{(1)}_{1,N-2} - (\alpha\lambda_E+\mu)p^{(1)}_{1,N-1} = 0.$$

which can be written in a recursive form

$$p^{(1)}_{1,0} = \frac{1}{\lambda_E+\mu} \frac{1-\rho_E}{1-\rho_E^{N+2}} \rho_E,$$

$$p^{(1)}_{1,j} = \frac{\lambda_E}{\lambda_E+\mu} p^{(1)}_{1,j-1} + \frac{1}{\lambda_E+\mu} \frac{1-\rho_E}{1-\rho_E^{N+2}} \rho_E^{j+1}, \quad j = 1, \ldots, N-2,$$

$$p^{(1)}_{1,N-1} = \frac{\lambda_E}{\alpha\lambda_E+\mu} p^{(1)}_{1,N-2} + \frac{1}{\alpha\lambda_E+\mu} \frac{1-\rho_E}{1-\rho_E^{N+2}} \rho_E^{N}.$$

The general solution for the first order non-homogeneous recurrence relation in $j = 0, \ldots, N-2$ is:

$$p^{(1)}_{1,j} = p^{(1)}_{1,0} \left(\frac{\lambda_E}{\lambda_E+\mu}\right)^j + \sum_{k=1}^{j} \frac{1}{\lambda_E+\mu} \frac{1-\rho_E}{1-\rho_E^{N+2}} \rho_E^{k+1} \left(\frac{\lambda_E}{\lambda_E+\mu}\right)^{j-k}$$

$$= \frac{1}{\lambda_E+\mu} \frac{1-\rho_E}{1-\rho_E^{N+2}} \rho_E \left(\frac{\lambda_E}{\lambda_E+\mu}\right)^j \sum_{k=0}^{j} \left(\frac{\lambda_E+\mu}{\mu}\right)^k$$

$$= \frac{1}{\lambda_E+\mu} \frac{1-\rho_E}{1-\rho_E^{N+2}} \left(\frac{\lambda_E}{\lambda_E+\mu}\right)^j \left(\left(\frac{\lambda_E+\mu}{\mu}\right)^{j+1} - 1\right), \quad j = 0, \ldots, N-2.$$

and

$$
\begin{aligned}
p^{(1)}_{1,N-1} &= \frac{\lambda_E}{\alpha\lambda_E+\mu}p^{(1)}_{1,N-2} + \frac{1}{\alpha\lambda_E+\mu}\frac{1-\rho_E}{1-\rho_E^{N+2}}\rho_E^N \\
&= \frac{1}{\alpha\lambda_E+\mu}\frac{1-\rho_E}{1-\rho_E^{N+2}}\left(\frac{\lambda_E}{\lambda_E+\mu}\right)^{N-1}\left(\left(\frac{\lambda_E+\mu}{\mu}\right)^{N-1}-1\right) + \frac{1}{\alpha\lambda_E+\mu}\frac{1-\rho_E}{1-\rho_E^{N+2}}\rho_E^N \\
&= \frac{1}{\alpha\lambda_E+\mu}\frac{1-\rho_E}{1-\rho_E^{N+2}}\left(\rho_E^{N-1}-\left(\frac{\lambda_E}{\lambda_E+\mu}\right)^{N-1}+\rho_E^N\right)
\end{aligned}
\tag{C.3}
$$

Now we have our first order approximation for $p_{1,N-1}$

$$p^{LT}_{1,N-1} = p^{(0)}_{1,N-1} + \lambda_S\, p^{(1)}_{1,N-1}$$

where $p^{(0)}_{1,N-1} = 0$ and $p^{(1)}_{1,N-1}$ is the expression (C.3).

Then, since the probability that N total packets are in the system will be composed just of the states $(0, N)$ and $(1, N-1)$ we obtain (6).

References

[1] S. Akhshabi, A.C. Begen, C. Dovrolis, An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP, in: Proceedings of the Second Annual ACM Conference on Multimedia Systems, MMSys '11, 2011, pp. 157–168.

[2] V. Jacobson, Congestion avoidance and control, in: Proceedings of ACM SIGCOMM, 1988, pp. 314–329.

[3] Cooperative Association for Internet Data Analysis, Analyzing UDP usage in Internet, http://www.caida.org/research/traffic-analysis/tcpudpratio/.

[4] K. Cho, K. Mitsuya, A. Kato, Traffic data repository at the WIDE project, USENIX 2000 FREENIX Track, 2000.

[5] P. Hurley, J.-Y.L. Boudec, M. Kara, P. Thiran, A novel scheduler for a low delay service within best-effort, in: L. Wolf, D. Hutchison, R. Steinmetz (Eds.), Quality of Service — IWQoS 2001: 9th International Workshop Karlsruhe, Germany, June 6–8, 2001 Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 389–403, http://dx.doi.org/10.1007/3-540-45512-4_30.

[6] M. Podlesny, S. Gorinsky, RD network services: differentiation through performance incentives, in: Proceedings of the ACM SIGCOMM 2008, in: SIGCOMM '08, ACM, 2008, pp. 255–266.

[7] L. Kleinrock, Queueing Systems, vol. 2, John Wiley and Sons, 1976.

[8] J. Cohen, The Single Server Queue, North-Holland, 1982.

[9] N. Jaiswal, Priority Queues, Academic Press, 1968.

[10] H. Takagi, Queueing Analysis: A Foundation of Performance Evaluation, Vol. 1: Vacation and Priority Systems, North-Holland, 1991.

[11] D. Wagner, U.R. Krieger, Analysis of a finite buffer with nonpreemptive priority scheduling, Comm. Statist. Stoch. Models 15 (1999) 345–365.

[12] J. Van Velthoven, B. Van Houdt, C. Blondia, The impact of buffer finiteness on the loss rate in a priority queueing system, Lecture Notes in Comput. Sci. 4054 (2006) 211–225.

[13] A. Bondi, An analysis of finite capacity queues with priority scheduling and common or reserved waiting areas, Comput. Oper. Res. 16 (1989) 217–233.

[14] D. Fiems, J. Walraevens, H. Bruneel, Performance of a partially shared priority buffer with correlated arrivals, in: Proc. of the 20th International Teletraffic Congress, ITC 20, Ottawa, Canada, 2007, pp. 582–593.

[15] T. Demoor, D. Fiems, J. Walraevens, H. Bruneel, Partially shared buffers with full or mixed priority, J. Ind. Manag. Optim. 7 (3) (2011) 735–751.

[16] K. Al-Begain, A. Dudin, V. Mushko, Novel queueing model for multimedia over downlink in 3.5G wireless network, in: Proc. of the 12th International Conference on Analytical and Stochastic Modelling Techniques and Applications, Riga, Latvia, 2005, pp. 111–117.

[17] X. Cheng, F. Akyildiz, A finite buffer two class queue with different scheduling and push-out schemes, in: IEEE Infocom, 1992, pp. 231–241.

[18] Y. Lee, B.D. Choi, B. Kim, D.K. Sung, Delay analysis of an $M/G/1/K$ priority queueing system with push-out scheme, Math. Probl. Eng. 2007 (2007) http://dx.doi.org/10.1155/2007/14504, Article ID 14504, 12 pages.

[19] A.S. Kapadia, M.F. Kazmi, A.C. Mitchell, Analysis of a finite capacity nonpreemptive priority queue, Comput. Oper. Res. 11 (1984) 337–343.

[20] Y. Lee, B.D. Choi, Queueing system with multiple delay and loss priorities for ATM networks, Inform. Sci. 138 (1) (2001) 7–29.

[21] K.E. Avrachenkov, G.L. Shevlyakov, N.O. Vilchevskii, Randomized push-out disciplines in priority queueing, J. Math. Sci. 122 (4) (2004).

[22] K.E. Avrachenkov, N.O. Vilchevsky, G.L. Shevlyakov, Priority queueing with finite buffer size and randomized push-out mechanism, Perform. Eval. 61 (1) (2005) 1–16, http://dx.doi.org/10.1016/j.peva.2004.08.006, http://www.sciencedirect.com/science/article/pii/S0166531604001063.

[23] M. Reiman, B. Simon, An interpolation approximation for queueing systems with Poisson input, Oper. Res. 36 (1988) 454–469.

[24] J. Walrand, Chapter 11 queueing networks, in: D. Heyman, M. Sobel (Eds.), Stochastic Models, in: Handbooks in Operations Research and Management Science, vol. 2, Elsevier, 1990, pp. 519–603, http://dx.doi.org/10.1016/S0927-0507(05)80175-6, http://www.sciencedirect.com/science/article/pii/S0927050705801756.

[25] K. Avrachenkov, J. Filar, P. Howlett, Analytic Perturbation Theory and its Applications, SIAM, 2013.