Analysis of the Early Packet Discard with per-VC Queueing Mechanism

K. Spaey

April 1999

Abstract

The Early Packet Discard (EPD) with per-VC Queueing mechanism is a packet discard scheme proposed in the literature to improve EPD performance in terms of fairness. This paper develops an analytical approximation model for evaluating the performance of the mechanism at a single node. As illustrated in the paper by numerical examples, the model allows the study of the evolution of performance measures such as buffer occupation, packet discard ratio and cell loss ratio when parameters of the input traffic or of the discarding mechanism are changed.

1 Introduction

The Unspecified Bit Rate (UBR) service category is intended for data applications that are not sensitive to delay and which generate bursty traffic that is difficult to characterize. Since no QoS commitments are made to connections of this service category, the network delivers a best effort service. It is however clear that when congestion in a network element occurs, each lost cell may belong to a different data packet. This leads to a low packet throughput (also called goodput), since losing a cell of a packet implies that the packet to which the cell belongs cannot be reassembled at the destination. Thus, once a cell of a packet is lost, all transmissions of subsequent cells of the packet are useless cell transmissions. Therefore, intelligent packet discarding mechanisms have been proposed and associated with the UBR service category, although they are not restricted to this service category only [3]. Packet discard mechanisms drop AAL5 packets through the use of the ATM User-to-User (AUU) bit in the Payload Type Identifier (PTI) field in the ATM header. Two well-known examples are Early Packet Discard (EPD) and Partial Packet Discard (PPD) [11]. EPD discards complete packets when a buffer threshold is exceeded, while PPD is triggered by buffer overflow and discards the remaining cells of packets from which a cell is lost.

Although the goodput may be increased considerably by using these packet discard schemes, they can not guarantee fairness [5, 10]. To avoid unfairness between competing flows, also per-flow information has to be incorporated into the mechanisms. Unfairness occurs because connections which lose packets will be forced by the transport protocol (e.g. TCP) to slow down their transmission rate, allowing other connections which did not loose packets to use more bandwidth and buffer space. Simulations show that schemes such as e.g. EPD with per-VC accounting, selective drop with per-VC accounting and EPD with per-VC queueing, alleviate the unfairness problem [5, 10]. The two schemes with per-VC accounting maintain fair buffer allocation at the time of cell discarding, while EPD with per-VC queueing also maintains fairness in the throughput as long as each VC has some cells in its queue.

*This report is available from ftp://wins.uia.ac.be/pub/pats/reports as 99-EPDperVC.ps.gz
In this report, we consider an analytical model for EPD with per-VC queueing. In the literature, a lot of exact analytical models are available for the EPD and PPD schemes [7, 9, 8]. A common factor in those papers is that the distinction between the normal mode of the switch in which arriving cells are admitted and the discarding mode in which arriving cells are discarded is made in the source models. There is thus a feedback between the state of the buffer and the source models. Since in EPD with per-VC queueing a round robin scheduler is used, constructing an exact model for this scheme will lead to an extremely large and thus analytically untractable model. Therefore, our analysis will follow an approximate approach, combining the source models that are used in the exact models for EPD and the approximate vacation models that are often used to model cyclic service systems [1, 4].

This report is organized as follows. In Section 2, we recall the EPD with per-VC queueing discarding scheme in detail. Section 3 introduces the model of the system, and Section 4 discusses the iterative algorithm to solve the system. Section 5 shows how performance measures can be obtained, while Section 6 presents a validation of the model and several numerical results. Section 7 concludes the report.

2 Early Packet Discard with per-VC Queueing

This section describes the EPD with per-VC queueing mechanism using the pseudocode of [10]. Cells from different VCs heading for the same output interface of an ATM switch will be buffered into different VC queues. In order to maintain fair buffer allocation, a round robin scheduler is employed to serve all VC queues. Denote $Q_j$ as the queue length for VC$_j$, $Q$ as the total queue size, $Th$ as the EPD threshold, $N$ as the number of active VCs (a VC is said to be active if it has at least one cell in its queue) and $Q_{max}$ as the maximum available buffer capacity. $K$ is a control parameter (typically $1 \leq K \leq 2$). In pseudocode, the algorithm then looks as follows:

When a cell of VC$_j$ arrives at the buffer:
- if the cell is the first cell of a packet
  - calculate $Th = \frac{K \times Q}{N}$
  - if $Q \geq Th$ and $Q_j \geq Th$
    - discard the cell
  - else
    - accept the cell into VC$_j$’s queue
    - $Q_j = Q_j + 1$
- else
  - if the first cell of the packet has been discarded
    - discard the cell
  - else
    - if $Q = Q_{max}$
      - discard the cell
    - else
      - accept the cell into VC$_j$’s queue
      - $Q_j = Q_j + 1$

Remark that whether a packet is discarded or not depends on two thresholds. If the buffer occupancy is below the fixed EPD threshold $Th$, no packets are discarded. If this threshold is exceeded, packet discarding is possible, depending on the second threshold $Th$, which is calculated as

$$Th = \frac{K \times Q}{N}.$$  \hspace{1cm} (1)

Since $Q$ and $N$ vary in time, also $Th$ varies in time. When $K = 1$, $Th$ represents the average buffer occupancy per VC. In the case that cells of a VC from which a new packet arrives already occupy more than a fair amount $Th$ of the buffer space, all the cells of the new packet are discarded.
3 Model Description

3.1 The System Model

Consider a discrete-time queueing system in which the time needed to transmit a cell, called a time slot, is chosen as time unit. The system consists of a buffer with maximum capacity $Q_{\text{max}}$ in which the traffic of $R + 1$ VCs arrives. One of these VCs is tagged, while the $R$ other VCs are considered as background VCs. Denote $Q_t(n)$ as the queue length of the tagged VC in time slot $n$ ($0 \leq Q_t(n) \leq Q_{\text{max}}$) and $Q(n)$ as the total queue size at time $n$ ($Q_t(n) \leq Q(n) \leq Q_{\text{max}}$).

3.2 The Input Process

Traffic for each of the VCs is generated by an on/off source, which changes its state from active to silent with probability $\alpha$, and from silent to active with probability $\beta$. When the source is active, a cell is generated in a time slot with probability $\lambda_{\text{on}}$. All cells which arrive during the same active period of the source constitute a packet. The packet length in number of cells is geometrically distributed with mean (see [7])

$$q = \frac{1 - (1 - \alpha)(1 - \lambda_{\text{on}})}{\alpha}$$

Because of the packet discard mechanism, not all cells generated by the sources will effectively enter the buffer. To model this, we define the states of each source similar as in [7]:

- off: the source does not generate a packet,
- on: the source is transmitting a packet of which the cells will not be discarded by the discarding mechanism (but they may be lost due to buffer overflow),
- on*: the source is transmitting a packet of which the cells will be discarded by the discarding mechanism.

So the source is active while it is in the on or on* state, but only the cells generated during an on period will effectively enter the buffer (except if they are lost due to buffer overflow).

We will suppose that all $R + 1$ sources are homogeneous in terms of traffic characteristics, and call the source which corresponds with the tagged VC the tagged source, the other $R$ sources are called background sources.

The state of the input process at the $n$-th slot can then completely be described by

- $S_{\text{on}}(n)$: the number of background sources in the on state,
- $S_{\text{on}*}(n)$: the number of background sources in the on* state,
- $S_t(n)$: the state of the tagged source,

with $0 \leq S_{\text{on}}(n) \leq R$, $0 \leq S_{\text{on}^*}(n) \leq R - S_{\text{on}}(n)$, $S_t(n) \in \{\text{on}, \text{on}^*, \text{off}\}$.

To describe the cell arrival probability at the switch at the $n$-th slot, define $a_{(j,k,l)}(r,s)$ as

$a_{(j,k,l)}(r,s)$ is the probability that $r$ cells ($r \in \{0, \ldots, R\}$) which originate from the background sources and that $s$ cells ($s \in \{0, 1\}$) which originate from the tagged source arrive at the switch when $S_{\text{on}}(n) = j, S_{\text{on}^*}(n) = k$ and $S_t(n) = l$.

Then $a_{(j,k,l)}(r,s)$ can be determined as follows:

$$a_{(j,k,l)}(r,s) = \begin{cases} \binom{j}{r}(\lambda_{\text{on}})^{j-r}(1 - \lambda_{\text{on}})^r & \text{if } s = 0, j \geq r \text{ and } l \in \{\text{on}*\}, \\ \binom{j}{r}(\lambda_{\text{on}})^{j-r}(1 - \lambda_{\text{on}})^{r+1} & \text{if } s = 0, j \geq r \text{ and } l = \text{on}, \\ \binom{j}{r}(\lambda_{\text{on}})^{r+1}(1 - \lambda_{\text{on}})^{j-r} & \text{if } s = 1, j \geq r \text{ and } l = \text{on}, \\ 0 & \text{otherwise.} \end{cases}$$
Denote by $A(r, s)$ the matrix
\[
A(r, s) = \text{diag}(a_{(0,0,0,0)}(r, s), a_{(0,0,0,0,0)}(r, s), \ldots, a_{(0,0,0,0,0,0)}(r, s), a_{(0,0,0,0,0,0)}(r, s)).
\] (4)

We will now obtain the transition probabilities of the number of sources being in each state, conditioned on whether the thresholds of the discarding mechanism are exceeded or not.

Consider first the transitions where $i$ can make in slot $n$:

- starting from on: to on or off,
- starting from on*: to on* or off,
- starting from off:
  - if $Q(n) < Th$ or if $Q(n) \geq Th$ and $Q_l(n) < \overline{T_h}(n)$: to on or off,
  - if $Q(n) \geq Th$ and $Q_l(n) \geq \overline{T_h}(n)$: to on* or off.

Remark that transitions from on to on* and from on* to on are not possible since the source has to pass always via off before it can generate a new packet. Or stated otherwise, a packet generation corresponds with a transition from off to on or on*.

Define now $b_{(j,k,l)}^{(r,s,t)}$ as the following probability:
\[
b_{(j,k,l)}^{(r,s,t)} = P\{S_{cn}(n+1) = r, S_{cn}^*(n+1) = s, S_l(n+1) = t | S_{cn}(n) = j, S_{cn}^*(n) = k, S_l(n) = l \text{ and } Q(n) < Th\}. \] (5)

If $Q(n) < Th$, only sources which are already in on* can stay in on*. So $b_{(j,k,l)}^{(r,s,t)} = 0$ if $k < s$. If $k \geq s$ and if $q$ represents the number of background sources which stay in on, then $b_{(j,k,l)}^{(r,s,t)}$ is given by
\[
b_{(j,k,l)}^{(r,s,t)} = \left[ \sum_{q = \max\{0, r + j + k - R\}}^{\min\{j, r\}} \binom{j}{q} \binom{k}{R - j - k} \binom{R - r - q}{s} (1 - \alpha)^{q+s} \alpha^q \beta^{q+k} \right] Z. \] (6)

where
\[
Z = \begin{cases} 
1 - \alpha & \text{if } l = t \text{ and } t \in \{\text{on, on*}\}, \\
\alpha & \text{if } l \in \{\text{on, on*}\} \text{ and } t = \text{off}, \\
\beta & \text{if } l = \text{off} \text{ and } t = \text{on}, \\
1 - \beta & \text{if } l = \text{off}, \\
0 & \text{otherwise}.
\end{cases}
\] (7)

Analogous to $b_{(j,k,l)}^{(r,s,t)}$, define $b_{(j,k,l)}^{(0,d)}$ as
\[
b_{(j,k,l)}^{(0,d)} = P\{S_{cn}(n+1) = r, S_{cn}^*(n+1) = s, S_l(n+1) = t | S_{cn}(n) = j, S_{cn}^*(n) = k, S_l(n) = l, Q(n) \geq Th, Q_l(n) < \overline{T_h}(n) \text{ and } d \text{ of the } R \text{ background queues exceed } \overline{T_h}(n)\}. \] (8)

and $b_{(j,k,l)}^{(1,d)}$ as
\[
b_{(j,k,l)}^{(1,d)} = P\{S_{cn}(n+1) = r, S_{cn}^*(n+1) = s, S_l(n+1) = t | S_{cn}(n) = j, S_{cn}^*(n) = k, S_l(n) = l, Q(n) \geq Th, Q_l(n) \geq \overline{T_h}(n) \text{ and } d \text{ of the } R \text{ background queues exceed } \overline{T_h}(n)\}. \] (9)
If there are \( v \) background sources which make a transition from off to on*, and \( q \) represents the number of background sources which stay on, then

\[
b_{(j,k)}^{(p,d)}(r,s,t) = \left[ \min(j,r) \right] \left[ \min(d,s,R) \ j \ k \ r+q \right] \left( r - q + v \right) \left( \prod_{i=1}^{v} \frac{d - i + 1}{R - i + 1} \right)
\]

\[
\left( \prod_{i=1}^{q} \frac{R - d - i + 1}{R - v - i + 1} \right) \left( \frac{j}{q} \right) \left( \frac{k}{s - v} \right) \left( \frac{R - j - k}{R - j - r + q - v} \right)
\]

\[
(1 - \alpha)^{q+s} \alpha^{j+k} (1 - \beta)^{s+v} \times Z,
\]

where \( Z \) is as in (7) for \( p = 0 \), and for \( p = 1 \),

\[
Z = \begin{cases} 
1 - \alpha & \text{if } l = t \text{ and } l \in \{ \text{on, on}^* \}, \\
\alpha & \text{if } l \in \{ \text{on, on}^* \} \text{ and } t = \text{off}, \\
\beta & \text{if } l = \text{off} \text{ and } t = \text{on}^*, \\
1 - \beta & \text{if } l = t = \text{off}, \\
0 & \text{otherwise.}
\end{cases}
\]

Remark that the factor

\[
\left( \frac{r - q + v}{v} \right) \left( \prod_{i=1}^{v} \frac{d - i + 1}{R - i + 1} \right) \left( \prod_{i=1}^{q} \frac{R - d - i + 1}{R - v - i + 1} \right)
\]

in (10) represents the probability that \( v (v \leq d) \) of the \( r - q + v \) new packets which arrive at time \( n \) of the background VCs arrive for a VC which exceeds \( T\bar{h}(n) \).

If we denote \( B^* \) and \( B^{(p,d)} \) (\( p = 0,1 \)) as the matrices

\[
B^* = \begin{pmatrix}
\begin{array}{ccc}
\ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots \\
\end{array}
\end{pmatrix}
\]

Then (i) \( C^*(r,s) := A(r,s)B^* \), (ii) \( C^{(p,d)}(r,s) := A(r,s)B^{(p,d)} \) and (iii) \( C^{(1,d)}(r,s) := A(r,s)B^{(1,d)} \)

are the transition matrices which represent the probabilities that \( r \) cells which originate from the background sources and \( s \) cells which originate from the tagged source arrive at the switch, given that (i) \( Q < T\bar{h} \), (ii) \( Q \geq T\bar{h} \), \( Q_t < T\bar{h} \) and \( d \) of the \( R \) background queues exceed \( T\bar{h} \), or (iii) \( Q \geq T\bar{h} \), \( Q_t \geq T\bar{h} \) and \( d \) of the \( R \) background queues exceed \( T\bar{h} \).

### 3.3 The Service Process

The queues are served according to a non-exhaustive (limited to one) cyclic service strategy, where one cell (if present) is served at each scanning epoch of a queue. The service time equals one time slot. At the end of this service, the server moves to the next queue. The switch-over time is supposed to be zero.

In principle, we would have to derive the joint queue length distribution of this multi-queueing system. But as the state space may become very large, we make an approximate analysis for
each queue in the system separately by deriving the joint probability distribution of the length of this queue and the total queue length. We model each queue as a queueing system with repeated server vacations. The time between two consecutive scanning instants of a queue $k$ is called a $k$-cycle. As before, we tag a queue and refer to the other queues as background queues. If the tagged queue is not empty at a scanning instant, its first cell is served. If after this service not all background queues are empty, the server will be unavailable for the tagged queue during a random time $v_1$, with $v_{1,l} = P\{v_1 = l\}$ ($l \in \{1, \ldots, R\}$), to serve one cell in each of the other non-empty queues of the system. Let $v_{2,l} = P\{v_2 = l\}$, where $l \in \{1, \ldots, R\}$. So the cycle of the tagged queue consists in this case of a service followed by a vacation. If after the service of the tagged queue all background queues are empty, the cycle consists of only one slot, namely that of the service. If the tagged queue is empty at a scanning instant, but the system is not, the server will be unavailable for the tagged queue during a random time $v_2$, with $v_{2,l} = P\{v_2 = l\}$ ($l \in \{1, \ldots, R\}$), to serve one cell in each of the non-empty background queues. In this case, the length of the cycle of the tagged queue equals the length of the vacation. If the system is completely empty at the scanning instant, the cycle of the tagged queue will be exactly one slot, after which the server will check the system again, to see if new cells have arrived meanwhile.

Remark that during a vacation, the system may not enter a state from which a service of the background queues is impossible, i.e. a state in which there are no cells in the background queues. But this implies that at all but the last slot of the vacation, the system may not be in a state where there is only one cell in the background queues and no background source in the on state, since if this would be the case, then at the next slot the system would be in one of the states we excluded before. On its turn, this restriction implies that at all but the last and last but one slot of the vacation, the system may not enter a state where there are two cells in the background queues and all background sources are in state on*, since a source has always to pass from on* via off before it can possibly go to on, and thus the system again would enter at the next slot a state of the excluded type.

### 3.4 The Queue Length Distribution at Scanning Instants

We describe the state of our approximate system at time $n$ by the discrete-time Markov chain with states $(Q_t(n), Q(n), S_{cn}(n), S_{cn'}(n), S_t(n))$. Denote the state space of this chain by $\mathcal{A}$. Denote further $\mathcal{B} := \{(r, s, j, k, l) \in \mathcal{A}|r = s\}$, $\mathcal{C} := \{(r, s, j, k, l) \in \mathcal{A}|s = r + 1 \text{ and } j = 0\}$, $\mathcal{D} := \{(r, s, j, k, l) \in \mathcal{A}|s = r + 2 \text{ and } k = R\}$ and $\mathcal{E} := \{(r, s, j, k, l) \in \mathcal{A}|r \neq 0\}$. To construct the evolution of this process between two scanning instants, we will first describe how this process evolves from slot to slot, by considering all different types of time slots which exist. A time slot can fall

- during a vacation of the tagged queue,
  - as the last slot of the vacation,
  - as the last but one slot of the vacation,
  - as the third-last slot of the vacation,
  - as a slot which is not of the type mentioned above,

- during a service of the tagged queue,

- during an idle period of the server because the system is totally empty.

To describe the evolution of the system from slot $n$ to slot $n + 1$, we have to be able to calculate $\mathcal{T}_t(n)$ at every slot, and to decide how many of the background queues exceed this $\mathcal{T}_t(n)$. But since the system state only provides the queue length of the tagged queue and of the total queue, we have to approximate the number of queues which are non-empty at time $n$, and the number of background queues which exceed $\mathcal{T}_t(n)$. Define $p_i(u, v, j, k, l)$ and $q_i(u, v, i, j, k, l)$ as $p_i(u, v, j, k, l)$ is the probability that there are $i$ VCs with a non-empty queue when $Q_t = u$, $Q = v$, $S_{cn} = j$, $S_{cn'} = k$ and $S_t = l$. 


and 

\[ q_d(u,v,i,j,k,l) \] is the probability that \( d \) of the background queues exceed \( TT \) when \( Q_t = u, Q = v, S_{cn} = j, S_{cn'} = k, S_I = l \) and there are \( i \) VCs with a non-empty queue.

Then denote by \( P_i(u,v) \) and \( Q_d(u,v,i) \) the matrices

\[
P_i(u,v) = \text{diag}(p_i(u,v,0,0,0), \ldots, p_i(u,v,R,0,0)),
\]

and

\[
Q_d(u,v,i) = \text{diag}(q_d(u,v,i,0,0,0), \ldots, q_d(u,v,i,R,0,0)).
\]

We will postpone the discussion of how to determine \( p_i(u,v,j,k,l) \) and \( q_d(u,v,i,j,k,l) \) to Section 4.

Consider first a slot which is the last one of a vacation of the tagged queue, and let \( V_1 \) be the transition matrix which describes the evolution of the system during that slot. \( V_1 \) is then a matrix with state space \((A \setminus B) \times A\) and consists of blocks \( V_{(Q_t(u),Q(u),(n+1),Q(n+1))} \) where \( V_{(u,v)}(y,x) \) with \( y \in \{u,u+1\} \) and with \( x \in \{v-u-1+y, \ldots, \min\{v-u-1+R+y,Q_{\text{max}}\}\} \) is given by

- \( v < Th, y = u \) and \( x \neq Q_{\text{max}}: C^*(x-v+1,0), \)
- \( v < Th, y = u+1 \) and \( x \neq Q_{\text{max}}: C^*(x-v,1), \)
- \( v < Th, y = u \) and \( x = Q_{\text{max}}: \)

\[
\sum_{i=0}^{R} \frac{1}{Q_{\text{max}} - v + 1 + i} \begin{cases} 
C^*(Q_{\text{max}} - v + 1 + i,0) \\
Q_{\text{max}} - v + 1 + i + 2 
\end{cases}
\]

- \( v \geq Th, y = u+1 \) and \( x \neq Q_{\text{max}}: \)

\[
\sum_{i=0}^{R+1} P_i(u,v) \sum_{d=0}^{R} Q_d(u,v,i) \left[ C^{(0,d)}(x-v+1,0)I_{\{u<\bar{x}\}} + C^{(1,d)}(x-v+1,0)I_{\{u>\bar{x}\}} \right];
\]

- \( v \geq Th, y = u+1 \) and \( x = Q_{\text{max}}: \)

\[
\sum_{i=0}^{R+1} P_i(u,v) \sum_{d=0}^{R} Q_d(u,v,i) \left[ C^{(0,d)}(Q_{\text{max}} - v + 1 + j,0)I_{\{u<\bar{x}\}} + C^{(1,d)}(Q_{\text{max}} - v + 1 + j,0)I_{\{u>\bar{x}\}} \right]
\]

\[
+ \frac{j+1}{Q_{\text{max}} - v + 2 + j} \left[ C^{(0,d)}(Q_{\text{max}} - v + 1 + j,1)I_{\{u<\bar{x}\}} + C^{(1,d)}(Q_{\text{max}} - v + 1 + j,1)I_{\{u>\bar{x}\}} \right];
\]
• if \( v \geq Th, y = u + 1 \) and \( x = Q_{\text{max}} \):

\[
\sum_{i=1}^{R+1} P_i(u,v) \sum_{d=0}^{R} Q_d(u,v,i) \sum_{j=0}^{R} \frac{Q_{\text{max}} - v + 1}{Q_{\text{max}} - v + 1 + j} C^{(y,d)}(Q_{\text{max}} - v + j,1) I_{\{u \leq x\}} + C^{(1,d)}(Q_{\text{max}} - v + j,1) I_{\{u > x\}}.
\]

(17)

All other blocks of \( V_1 \) are 0. Remark that in the formulas above, we assumed that all cells have equal chance to get lost due to buffer overflow.

For a slot which is the last but one of a vacation of the tagged queue, let \( V_2 \) be the transition matrix which describes the evolution of the system during that slot. \( V_2 \) can be obtained from \( V_1 \) in the following way: if \( W_2 \) is the submatrix of \( V_1 \) which corresponds with the state space \((\mathcal{A} \setminus (\mathcal{B} \cup \mathcal{C})) \times (\mathcal{A} \setminus \mathcal{B})\), then

\[
V_2 = \text{diag}(1/(W_2e))W_2,
\]

(18)

where \((1/x)\) denotes the element-by-element inverse of vector \(x\), and \(e\) denotes a column vector of 1's.

For a slot which is the third-last slot of a vacation of the tagged queue, we define \( V_3 \) as the transition matrix of the evolution of the system during that slot. \( V_3 \) is calculated from \( V_1 \) by defining \( W_3 \) as the submatrix of \( V_1 \) which corresponds with the state space \((\mathcal{A} \setminus (\mathcal{B} \cup \mathcal{C} \cup \mathcal{D})) \times (\mathcal{A} \setminus (\mathcal{B} \cup \mathcal{C} \cup \mathcal{D}))\). Then

\[
V_3 = \text{diag}(1/(W_3e))W_3.
\]

(19)

Finally, we define \( V_4 \) as the transition matrix which describes the evolution of the system during a slot of a vacation of the tagged queue which is not the last, nor the last but one or the third-last of that vacation. If \( W_4 \) is the submatrix of \( V_1 \) which corresponds with the state space \((\mathcal{A} \setminus (\mathcal{B} \cup \mathcal{C} \cup \mathcal{D})) \times (\mathcal{A} \setminus (\mathcal{B} \cup \mathcal{C} \cup \mathcal{D}))\), then

\[
V_4 = \text{diag}(1/(W_4e))W_4.
\]

(20)

Consider now a slot in which the tagged queue is served. Let \( S \) be the transition matrix which describes the evolution of the system during that slot. \( S \) is then a matrix with state space \( \mathcal{E} \times \mathcal{A} \), and consists of blocks \( S_{(Q_1,n)} \), where \( S_{(u,v)} \) with \( y \in \{u - 1, u\} \) and \( x \in \{v - u + y, \ldots, \min\{v - u + R + y, Q_{\text{max}}\}\} \) is obtained by applying the same rules as for the construction of \( V_1 \) \((17)\), but with everywhere \( y \) replaced by \( y + 1 \). Remark that for \( u \neq v \), this implies that \( S_{(u,v)} = V_{(u,v)}(y+1,x) \). For \( u = v \) however, this is not true, since \( V_{(u,v)}(y+1,x) \) is not defined.

For a slot during an idle period of the server, let \( V_0 \) describe the evolution of the system. Then \( V_0 \) is a matrix with state space \( \{(r, s, j, k, l) \in \mathcal{A} | r = s = 0\} \times \mathcal{A} \) which consists of blocks \( V_{(Q_{(n+1)} \cap (Q_{(n+1)} \cup Q_{(n+1)} \cup Q_{(n+1)}))} \), where \( V_{(0,0)} \) with \( y \in \{0, 1\} \) and \( x \in \{y, \ldots, \min(y + R, Q_{\text{max}})\} \) is given by

- if \( y = 0 \) and \( x \neq Q_{\text{max}} \): \( C^*(x,0) \),
- if \( y = 1 \) and \( x \neq Q_{\text{max}} \): \( C^*(x-1,1) \),
- if \( y = 0 \) and \( x = Q_{\text{max}} \): \( \sum_{i=0}^{R} Q_{\text{max}} C^*(Q_{\text{max}} + i,0) + \sum_{i=0}^{R} Q_{\text{max}} C^*(Q_{\text{max}} + i,1) \),
- if \( y = 1 \) and \( x = Q_{\text{max}} \): \( \sum_{i=0}^{R} Q_{\text{max}} C^*(Q_{\text{max}} - 1 + i,1) \).
All other blocks of $V_C$ equal 0.

Some last notational conventions we have to introduce before we can consider the system at scanning instants are notations for submatrices of already defined matrices. Denote with $S_1, \ldots, S_4$ the submatrices of $S$ which correspond with the state space $E \times B$ for $S_1$, $E \times C$ for $S_2$, $E \times D$ for $S_3$, and $E \times (A \setminus (B \cup C \cup D))$ for $S_4$. Further, $V^{(1)}$, $V^{(2)}$, and $V^{(3)}$ are submatrices of $V_i$ which correspond with state space $(A \setminus (B \cup C \cup D)) \times A$ for $V^{(1)}$, $D \times A$ for $V^{(2)}$, and $C \times A$ for $V^{(3)}$, and $V^{(1)}$ and $V^{(2)}$ are submatrices of $V_2$ which correspond with state space $(A \setminus (B \cup C \cup D)) \times (A \setminus B)$ for $V^{(1)}$ and $D \times (A \setminus B)$ for $V^{(2)}$.

Now consider the system at scanning instants. The evolution of the system between two scanning instants is described by the transition matrix $Q$, with $A \times A$ as state space. The elements $Q_{(r,s,j,k,l)(r',s',j',k',l')}$ are given by the corresponding elements of the following matrices:

- if $(r,s,j,k,l) \in B$ with $r = 0$: $V_0$,
- if $(r,s,j,k,l) \in C$ with $r = 0$: $V_1$,
- if $(r,s,j,k,l) \in D$ with $r = 0$: \( \frac{v_{1,1}}{v_{1,1} + v_{1,2}} V_1 + \frac{v_{1,2}}{v_{1,1} + v_{1,2}} V_2 V_1 \),
- if $(r,s,j,k,l) \in A \setminus (B \cup C \cup D)$ with $r = 0$: \( v_{1,1} V_1 + v_{1,2} V_2 V_1 + \sum_{l=1}^{\max} v_{1,l}(V_4)^l 3 V_3 V_2 V_1 \),
- if $(r,s,j,k,l) \in E$ with $r' = s'$:
  \[ S_1 + S_2 V^{(3)}_1 + \frac{v_{2,2}}{v_{2,1} + v_{2,2}} S_3 V^{(2)}_2 V_1 + v_{2,1} S_4 V^{(1)}_1 + v_{2,2} S_4 V^{(1)}_2 V_1 + \sum_{l=3}^{\max} v_{2,l} S_4 (V_4)^l 3 V_3 V_2 V_1, \]
- if $(r,s,j,k,l) \in E$ with $r' \neq s'$:
  \[ \frac{v_{2,1}}{v_{2,1} + v_{2,2}} S_3 V^{(2)}_1 + v_{2,1} S_4 V^{(1)}_1 + v_{2,2} S_4 V^{(1)}_2 V_1 + \sum_{l=3}^{\max} v_{2,l} S_4 (V_4)^l 3 V_3 V_2 V_1. \]

If we divide $Q$ into blocks $Q_{i,j}$ ($i,j \in \{0, \ldots, \max\}$), where $i$ and $j$ correspond with the number of customers in the tagged queue, then the blocks $Q_{i,j}$ with $j < i - 1$ are 0, since at most one cell can leave the tagged queue between two scanning instants. Also the blocks $Q_{i,j}$ with $i > R$ and the blocks $Q_{i,j}$ with $i > R$ and $j > R + i$ are 0, since a vacation of the tagged queue lasts maximum $R$ slots, so at most $R$ cells can arrive in the tagged queue when $i = 0$, and at most $R + 1$ cells can arrive in the tagged queue when $i > 0$. But in the last case, one cell will leave the system during service of the tagged queue, resulting in a maximum net growth of $R$ cells. Remark that only the diagonal blocks $Q_{i,i}$ of $Q$ are square, and they are becoming smaller for increasing $i$. Denote the stationary distribution of $Q$ by $q$, i.e.

\[ q Q = q \quad \text{and} \quad q e = 1. \]

Because of the upper block-Hessenberg structure of $Q$, we can find $q$ by applying for example the algorithm of Grassmann et al. [6] extended to block partitioned matrices as explained in [2]. This algorithm has no problems dealing with the non-equal sized blocks $Q_{i,j}$.

### 3.5 The Queue Length Distribution at an Arbitrary Instant

From the stationary joint distribution $q$ of the queue length of the tagged queue and the queue length of the total queue at scanning instants, the same joint distribution $z$ at an arbitrary instant
is derived. Choose \(n\) an arbitrary time instant and \(a \in A\) an arbitrary state of the system. It is clear that \(n\) will always fall in a time interval defined by two successive scanning instants \(s\) and \(s', i.e. n \in [s, s']\). Denote by \(L\) the length of this time interval, and denote the state of the system at time \(n\) by \(U(n)\). Then the probability that \(U(n) = a\) is calculated by

\[
P\{U(n) = a\} = \sum_{l=1}^{R+1} \sum_{b \in A} \sum_{k=1}^{l} P\{U(n) = a, U(s) = b, L = l \text{ and } n = s + k\}
\]

\[
= \sum_{l=1}^{R} \sum_{b \in \{0, \ldots, a, a', a''\} \in A} \sum_{k=1}^{l} P\{U(n) = a, U(s) = b, L = l \text{ and } n = s + k\}
\]

\[
+ \sum_{l=1}^{R+1} \sum_{(r, l, u, v, w) \in E} \sum_{k=1}^{l} P\{U(n) = a, U(s) = b, L = l, n = s + k \text{ and } U(s + 1) = c\}
\]

\[
= \sum_{l=1}^{R} \sum_{b \in \{0, \ldots, a, a', a''\} \in A} \sum_{k=1}^{l} P\{U(n) = a, U(s) = b, L = l \text{ and } n = s + k\}
\]

\[
P\{n = s + k|U(s) = b \text{ and } L = l\} \cdot P\{L = l|U(s) = b\} \cdot P\{U(s) = b\}
\]

\[
+ \sum_{l=1}^{R+1} \sum_{(r, l, u, v, w) \in E} \sum_{k=1}^{l} P\{U(n) = a, U(s) = b, L = l, n = s + k \text{ and } U(s + 1) = c\}
\]

\[
P\{n = s + k|U(s) = b, L = l \text{ and } U(s + 1) = c\} \cdot P\{L = l|U(s) = b\} \cdot P\{U(s + 1) = c|U(s) = b\} \cdot P\{U(s) = b\}
\]

(23)

The different parts in this formula are calculated as follows:

- The probabilities \(P\{U(n) = a|U(s) = b, L = l \text{ and } n = s + k\}\) and \(P\{U(n) = a|U(s) = b, L = l, n = s + k \text{ and } U(s + 1) = c\}\) are obtained from the formulas in (21), which describe the evolution of the system between two scanning instants.

- The probabilities \(P\{n = s + k|U(s) = b \text{ and } L = l\}\) and \(P\{n = s + k|U(s) = b, L = l \text{ and } U(s + 1) = c\}\) equal \(\frac{1}{l}\), since if an arbitrary instant falls in an interval of length \(l\), it has equal probability to fall in any slot in this interval.

- The probabilities \(P\{L = l|U(s) = b\}\) and \(P\{L = l|U(s) = b \text{ and } U(s + 1) = c\}\) equal

\[
P\{L = l|U(s) = b\} = \frac{\text{Length of an interval } [s, s'][U(s) = b]}{E| \text{length of an interval } [s, s'][U(s) = b]} \quad (24)
\]

\[
P\{L = l|U(s) = b\} = \frac{\text{Length of an interval } [s, s'][U(s) = b \text{ and } U(s + 1) = c]}{E| \text{length of an interval } [s, s'][U(s) = b \text{ and } U(s + 1) = c]} \quad (25)
\]

Remark that the probabilities and means in (24) and (25) can be read from (21), and depend on the state \(b\) in which the system is at the first scanning instant of an interval.

- The probabilities \(P\{U(s + 1) = c|U(s) = b\}\) is given by \((S_i)_{b,c}\), with \(i = 1, 2, 3\) or \(4\), depending on if \(c \in B, C, D\) or \(A \setminus (B \cup C \cup D)\).

- The probability \(P\{U(s) = b\}\) equals the probability that the system is in state \(b\) at a scanning instant, and thus equals \(q_b\).
This results in

\[
P\{U(n) = a\} = \sum_{b=\{0,0,m,n,p\}\in B} (V_0)_{b,a} q_b + \sum_{b=\{0,1,0,n,p\}\in C} (V_1)_{b,a} q_b
\]

\[
+ \sum_{b=\{0,2,0,p\}\in D} \left( \frac{v_{1,1}}{v_{1,1} + 2v_{1,2}} (V_1)_{b,a} + \frac{v_{1,2}}{v_{1,1} + 2v_{1,2}} ((V_2)_{b,a} 1_{\{a\in A\setminus B\}} + (V_2 V_1)_{b,a}) \right) q_b
\]

\[
+ \sum_{b=\{0,1,0,p\}\in A \setminus (B \cup C \cup D)} \left[ \frac{1}{E[t_1]} \left( v_{1,1}(V_1)_{b,a} + v_{1,2} ((V_2)_{b,a} 1_{\{a\in A\setminus B\}} + (V_2 V_1)_{b,a}) \right) q_b \right]
\]

\[
+ \sum_{l=3} R \left( \frac{v_{1,1}}{l(l-1)} \left( \left( V_4 \right)^l b,a_{1_{\{a\in A\setminus B\}}} + \left( V_4 \right)^l 3 V_3 V_2 V_1 b,a \right) \right) q_b
\]

\[
+ \sum_{b=\{r,s,t,k,l\}\in E} \left( \left( S_1 \right)_{b,a} + \frac{1}{2} \left( (S_2)_{b,a} + (S_2 V_1^{(3)})_{b,a} \right) + \frac{v_{2,1}}{2v_{2,1} + 3v_{2,2}} ((S_3)_{b,a} + (S_3 V_1^{(2)})_{b,a}) \right)
\]

\[
+ \frac{1}{E[t_2]} \left[ \frac{v_{2,1}}{l(l-2)} \left( (S_4)_{b,a} + (S_4 V_1^{(1)})_{b,a} \right) + v_{2,2} \left( (S_4)_{b,a} + (S_2 V_2^{(2)})_{b,a} + (S_4 V_2^{(1)})_{b,a} \right) \right]
\]

\[
+ \sum_{l=3} R \left( \left( S_4 \right)_{b,a} + \sum_{k=1}^3 \left( S_4 (V_4)^k \right)_{b,a} + \left( S_4 (V_4)^l 3 V_3 V_2 b,a \right) + \left( S_4 (V_4)^l 3 V_3 V_2 V_1 b,a \right) \right) q_b.
\]  

(26)

4 Calculating the Queue Length Distributions

In Section 3 we have described how to model the multi-queueing system by considering one tagged queue together with the total queue. The following algorithm describes how \( q \), resp. \( z \), the stationary joint distribution of the queue length of the tagged and the queue length of the total queue can be computed at a scanning instant of the tagged queue, resp. at an arbitrary instant. Define \( \phi \) and \( \psi \) as

\[
\phi, \text{ resp. } \psi, \text{ is the stationary joint queue length distribution of the tagged and the total queue at scanning instants of the tagged queue during a cycle of a particular background queue, knowing that this background queue was empty, resp. non-empty at its scanning instant,}
\]

and define \( \hat{\phi} \) and \( \hat{\psi} \) as

\[
\hat{\phi}, \text{ resp. } \hat{\psi}, \text{ is the stationary joint queue length distribution of the tagged and the total queue at an arbitrary instant, knowing that during a cycle of a particular background queue, this background queue was empty, resp. non-empty at its scanning instant.}
\]

We will calculate these stationary distributions in an iterative way: in every iteration step we assume we have these distributions available (the ‘old’ distributions), we use them to calculate all unknowns (vacation distribution, \( p(u,v,j,k,l) \) and \( q(u,v,i,j,k,l) \)) needed to construct \( Q \) as explained in Section 3, and from \( Q \) and the vacation distributions we obtain the ‘new’ stationary distributions. This results in the following algorithm:

1. Assign initial distributions to \( \phi, \psi, \hat{\phi} \) and \( \hat{\psi} \).
2. Calculate
(a) the vacation distributions $v_3, v_4, v_5$ and $v_6$ using $\phi$ and $\psi$,
(b) all the values $p_l(u, v, j, k, l)$ and $q_d(u, v, i, j, k, l)$ using $\tilde{\phi}$ and $\tilde{\psi}$.

3. Compose $Q$ as described in Section 3, and calculate its stationary distribution $q$.
(a) using $v_1 := v_3$ and $v_2 := v_4$, resulting in $\phi := q$.
(b) using $v_1 := v_5$ and $v_2 := v_6$, resulting in $\psi := q$.

4. Compute the stationary distribution $z$ at arbitrary instants.
(a) using $v_1 := v_3$ and $v_2 := v_4$ and $\phi$ as the stationary distribution at scanning instants, resulting in $\tilde{\phi} := z$.
(b) using $v_1 := v_5$ and $v_2 := v_6$ and $\psi$ as the stationary distribution at scanning instants, resulting in $\tilde{\psi} := z$.

5. Repeat steps 2–4 until in two consecutive steps, the difference between the ‘old’ and the ‘new’ distribution $\phi$ and $\psi$ are element-wise less than a given $\epsilon$.

6. Calculate
(a) the vacation distributions $v_7$ and $v_8$ using $\phi$ and $\psi$,
(b) all values $p_l(u, v, j, k, l)$ and $q_d(u, v, i, j, k, l)$ using $\tilde{\phi}$ and $\tilde{\psi}$.

7. Compose $Q$ as described in Section 3, and calculate its stationary distribution $q$. Compute $z$ from $q$. Use $v_1 := v_7$ and $v_2 := v_8$.

The vacation distributions $v_3, \ldots, v_8$ used in the algorithm above are defined as follows:

$v_3, v_4, v_5$ and $v_6$ are the lengths of a vacation during a cycle of the tagged queue, conditioned on the facts that the tagged queue was empty (for $v_3$ and $v_5$) / non-empty (for $v_4$ and $v_6$) at the start of the cycle and that a particular background queue is empty (for $v_3$ and $v_4$) / non-empty (for $v_5$ and $v_6$) at its scanning instant during this cycle.

$v_7$ resp. $v_8$ is the length of a vacation during a cycle of the tagged queue, conditioned on the fact that the tagged queue was empty, resp. non-empty at the start of the cycle.

Define

$$
\phi_0 = \frac{\sum_{(c, s, j, k, l) \in A} \phi_{(c, s, j, k, l)}}{1 - \sum_{(c, s, j, k, l) \in B} \phi_{(c, s, j, k, l)}} \quad \text{and} \quad \psi_0 = \frac{\sum_{(c, s, j, k, l) \in A} \psi_{(c, s, j, k, l)}}{1 - \sum_{(c, s, j, k, l) \in B} \psi_{(c, s, j, k, l)}}
$$

(27)

Since all sources in our input process are homogeneous, the $z$-transforms $V_i(z)$ of the random variables $v_i, l \in \{3, \ldots, 8\}$ are given by:

$$
V_3(z) = \frac{(1 - \phi_0)z + \phi_0}{1 - (\phi_0)^R} \left( R - 1 \right) R^1 \left( R - 1 \right) \sum_{i=1}^{R} \frac{(R-1)}{i} (1 - \phi_0)^i (\phi_0)^R 1^1 i z^i.
$$

(28)

$$
V_4(z) = \frac{1}{1 - (\psi_0)^R} \sum_{i=1}^{R} \left( R - 1 \right) \left( R - 1 \right) \psi_0 (\psi_0)^R 1^1 i z^i.
$$

(29)

$$
V_5(z) = z \left( (1 - \phi_0)z + \phi_0 \right) R^1 \left( R - 1 \right) \sum_{i=1}^{R} \left( i \right) (1 - \phi_0)^i (\phi_0)^R 1^1 i z^i.
$$

(30)
\[ V_0(z) = \sum_{i=1}^{R} \frac{(R-1)}{i-1} (1-\psi)^i 1(\psi)^R z^i, \]  
(31)

\[ V_T(z) = \frac{((1-\phi)c+z+\phi c)^R - (\phi c)^R}{1-(\phi c)^R} = \frac{1}{1-(\phi c)^R} \sum_{i=1}^{R} \binom{R}{i} (1-\phi)^i (\phi c)^R z^i, \]  
(32)

\[ V_s(z) = \frac{1}{1-(\psi c)^R} \sum_{i=1}^{R} \binom{R}{i} (1-\psi)^i (\psi c)^R z^i. \]  
(33)

Remark that these formulas include the fact that the length of a vacation is always larger than zero, since the server will only take a vacation if not all the background queues are empty. Remark also that in the definitions of \( \phi \) and \( \psi \) the probabilities \( \phi_{[v,0,j,k,l]} \) and \( \psi_{[v,0,j,k,l]} \) are excluded, since if the system is completely empty, it will not take a vacation but it will be idle for a slot (see (21)).

We will now describe how the values \( p_i(u,v,j,k,l) \) and \( q_{d}(u,v,i,j,k,l) \) are calculated (\( i \in \{1, \ldots, R+1\} \) and \( d \in \{0, \ldots, R\} \)). First remark that if \( u = v \),

\[ p_i(u,v,j,k,l) = \begin{cases} 1 & \text{if } i = 1, \\ 0 & \text{otherwise}, \end{cases} \]  
(34)

and

\[ q_{d}(u,v,i,j,k,l) = \begin{cases} 1 & \text{if } i = 1 \text{ and } d = 0, \\ 0 & \text{otherwise}, \end{cases} \]  
(35)

since all cells in the system are located in the tagged queue. When \( u \neq v \), the number of background VCs which are non-empty lies between 1 and \( \min\{R, v-u\} \). Thus, if \( u \neq v \),

\[ p_i(u,v,j,k,l) = 0 \text{ if } i - 1 \{ u \neq 0 \} = 0 \text{ or } i - 1 \{ u \neq 0 \} > \min\{R, v-u\}. \]  
(36)

In the other cases, we calculate \( p_i(u,v,j,k,l) \) as follows: if \( u = 0 \), define \( X := \hat{\phi} \), otherwise define \( X := \hat{\psi} \). Construct a new probability distribution \( Y \) from \( X \) by only considering states which belong to

\[ \mathcal{F} = \{ (y,x,r,s,t) \in \mathcal{A} | y \leq v-u, x = v, r + 1 \{ t=cn \} = j + 1 \{ t=cn \}, s + 1 \{ t=cn \} = k + 1 \{ t=cn \} \}. \]  
(37)

and by giving them in \( Y \) the probabilities

\[ Y_{i(x,r,s,t)} := \frac{X_{(y,x,r,s,t)}}{\sum_{(y,x,r,s,t) \in \mathcal{F}} X_{(y,x,r,s,t)}}, \]  
(38)

Then define \( \forall i \in \{0, \ldots, v-u\} \):

\[ s_i := \sum_{(i,x,r,s,t) \in \mathcal{F}} Y_{i(x,r,s,t)}. \]  
(39)

Then \( (s_0, \ldots, s_{v-u}) \) is a probability distribution for the occupation of a particular background queue, given that the system is in state \( (u,v,j,k,l) \). Remark that we consider in \( \mathcal{F} \) only states with \( y \leq v - u \) as first component, since we know that in the particular background queue no more than \( v-u \) cells can be present.

The \( z \)-transform of the number of VCs with non-empty queue is then given by \( P(z) \). Since all sources are homogeneous, \( P(z) \) equals

13
• if $u = 0$:

$$P(z) = \frac{(1 - s_0)z + s_0)^M}{1 - (s_0)^M} = \frac{1}{1 - (s_0)^M} \sum_{i=1}^{M} \left( \frac{M}{i} \right) (1 - s_0)^{i} (s_0)^{M - i} z^i,$$

(40)

• if $u \neq 0$:

$$P(z) = z \frac{\left( (1 - s_0)z + s_0)^M - (s_0)^M \right)}{1 - (s_0)^M} = \frac{1}{1 - (s_0)^M} \sum_{i=2}^{M+1} \left( \frac{M}{i - 1} \right) (1 - s_0)^{i-1} (s_0)^{M + 1} z^i,$$

(41)

where $M = \min \{ v - u, R \}$, the maximum number of non-empty background queues. Remark that the formulas above include the fact that not all background queues can be empty, since $u \neq v$. We will approximate $q_d(u, v, i, j, k, l)$ as follows: consider all possibilities $(n_1, \ldots, n_k)$ to spread $v - u$ cells among $k$ background queues, $k = i - 1 \{uv=0\}$, and assign to them a probability

$$P(n_1, \ldots, n_k) = \left( \sum_{\{n_1, \ldots, n_k\} \text{ such that } n_1 + \ldots + n_k = v - u} \prod_{j=1}^{k} s_{n_j} \right)^{1/k} \prod_{j=1}^{k} s_{n_j}.$$  

(42)

Then $q_d(u, v, i, j, k, l)$ is given by

$$q_d(u, v, i, j, k, l) = \sum_{\{n_1, \ldots, n_k\} \text{ such that } \{n_1, \ldots, n_k\} \geq \{s_0^M\} = d} P(n_1, \ldots, n_k).$$

(43)

5 Performance Measures

Important performance measures we can calculate from the obtained queue length distributions are the packet discard ratio and the cell loss ratio.

5.1 The Packet Discard Ratio

The packet discard ratio is defined as

$$P_{\text{disc}} = \frac{\text{mean number of packets discarded by the discarding mechanism in a time slot}}{\text{mean number of packets generated in a time slot}}.$$  

(44)

Since all our sources are homogeneous, $P_{\text{disc}}$ is also given by

$$P_{\text{disc}} = \frac{P\{ \text{a packet from the tagged source is discarded in a time slot} \}}{P\{ \text{the tagged source generates a packet in a time slot} \}}.$$  

(45)

The probability that the tagged source generates a new packet in a time slot is given by the probability that the tagged source makes a transition from the off state to the on or on* state, and while the source is in this on or on* state, it generates at least one cell. Denote by $I$ the probability that the tagged source makes a transition from off to on or on*, and with $P_0$ the
probability that no cell is generated during an on or on* period. \( I \) is given by

\[
I = \sum_{(u,v,j,k,\text{off}) \in A} P\{\text{the tagged source makes a transition from off to on or on* | } \\
\quad \text{the system is in state (u, v, j, k, off)}\} P\{\text{the system is in state (u, v, j, k, off)}\}
\]

\[
= \sum_{(u,v,j,k,\text{off}) \in A} \beta z_{(u,v,j,k,\text{off})}.
\]

(46)

If a packet from the tagged source is discarded in a time slot, the tagged source makes a transition from off to on* in that time slot. Denote by \( J \) the probability that the tagged source makes a transition from off to on*. Since a transition from off to on* is the only type of transition a source can make to arrive in the on* state, and since a source has to leave the on* state by making a transition to the off state, \( J \) also equals the probability that a source makes a transition from on* to off in a time slot. This implies that

\[
J = \sum_{(u,v,j,k,\text{on*}) \in A} P\{\text{the tagged source makes a transition from on* to off | } \\
\quad \text{the system is in state (u, v, j, k, on*)}\} P\{\text{the system is in state (u, v, j, k, on*)}\}
\]

\[
= \sum_{(u,v,j,k,\text{on*}) \in A} \alpha z_{(u,v,j,k,\text{on*})}.
\]

(47)

So \( P_{\text{disc}} \) equals

\[
P_{\text{disc}} = \frac{(1 - P_c) J}{(1 - P_c) I} = \frac{\alpha \sum_{(u,v,j,k,\text{on*}) \in A} z_{(u,v,j,k,\text{on*})}}{\beta \sum_{(u,v,j,k,\text{off}) \in A} z_{(u,v,j,k,\text{off})}}.
\]

(48)

5.2 The Cell Loss Ratio

The cell loss ratio is defined as

\[
\text{CLR} = \frac{\text{mean number of cells which are lost due to buffer overflow in a time slot}}{\text{mean number of cells which arrive at the buffer in a time slot}}.
\]

(49)

Because of the homogeneity of all input sources, we can calculate the CLR also by

\[
\text{CLR} = \frac{P\{\text{a cell of the tagged source is lost due to buffer overflow in a time slot}\}}{P\{\text{a cell of the tagged source arrives at the buffer in a time slot}\}}.
\]

(50)

Denote the numerator of this fraction by \( L \), the denominator by \( M \). Since a cell of the tagged source can only arrive at the buffer if the source is in the on state, \( M \) is given by

\[
M = \lambda_{cn} \sum_{(u,v,j,k,\text{cn}) \in A} z_{(u,v,j,k,\text{cn})}.
\]

(51)

\( L \) is obtained as follows:

\[
L = \sum_{w=1}^{R+1} \sum_{(u,v,j,k,\text{cn}) \in A} P\{\text{a cell of the tagged source is lost due to buffer overflow in } \\
\quad \text{a time slot, and in total w cells are lost due to buffer overflow in that time slot | } \\
\quad \text{the system is in state (u, v, j, k, on)}\} P\{\text{the system is in state (u, v, j, k, on)}\}.
\]

(52)
If there are \( j \) background sources in the on state, from which \( l \) generate a cell, then \( w \) cells will be lost due to buffer overflow, with \( w = v + l - Q_{\text{max}} \) in the case that the system is not empty, and \( w = l + 1 - Q_{\text{max}} \) otherwise. This gives:

\[
L = \sum_{w=1}^{R+1} \sum_{i=1}^{Q_{\text{max}}} \sum_{j=w}^{R} \sum_{(u,v,j,k,\text{on}) \in A} P\{\text{the system is in state } (u,v,j,k,\text{on})\}
\]

\[
\lambda_{\text{cn}} \frac{w}{w - v + Q_{\text{max}} + 1} \left( \frac{j}{w - v + Q_{\text{max}}} \right)^w v + Q_{\text{max}} (1 - \lambda_{\text{cn}})^j \left( w + v \right) Q_{\text{max}}
\]

\[
+ \sum_{w=1}^{R+1} \sum_{j=w}^{R} \sum_{(u,0,j,k,\text{on}) \in A} P\{\text{the system is in state } (u,0,j,k,\text{on})\}
\]

\[
\lambda_{\text{cn}} \frac{w}{w + Q_{\text{max}}} \left( \frac{j}{w - 1 + Q_{\text{max}}} \right)^j v + Q_{\text{max}} (1 - \lambda_{\text{cn}})^j \left( w + 1 \right) Q_{\text{max}}.
\]  \tag{53}

By changing the order of summation in \( L \), we then obtain that

\[
\text{CLR} = \left[ \sum_{i=1}^{Q_{\text{max}}} \sum_{j=1}^{R} \sum_{(u,v,j,k,\text{on}) \in A} z_{(u,v,j,k,\text{on})} \right] \left( \sum_{w=1}^{R} \left( \lambda_{\text{cn}} \frac{w}{w - v + Q_{\text{max}} + 1} \right)^j \left( w + v \right) Q_{\text{max}} \right)
\]

\[
\left( \frac{j}{w - v + Q_{\text{max}}} \right)^w v + Q_{\text{max}} + 1 \right)(1 - \lambda_{\text{cn}})^j \left( w + v \right) Q_{\text{max}} + \sum_{j=Q_{\text{max}}}^{R} \sum_{(u,0,j,k,\text{on}) \in A} z_{(u,0,j,k,\text{on})} \right] \left( \lambda_{\text{cn}} \frac{w}{w - 1 + Q_{\text{max}}} \right)^j v + Q_{\text{max}} (1 - \lambda_{\text{cn}})^j \left( w + 1 \right) Q_{\text{max}}
\]

\[
\left( \lambda_{\text{cn}} \sum_{(u,v,j,k,\text{on}) \in A} z_{(u,v,j,k,\text{on})} \right)^j. \tag{54}
\]

6 Model Validation and Numerical Results

To validate our model, results obtained by the analytical model are compared with simulation results. Figure 1 shows the distribution of the length of the total queue and of a VC queue for three examples with parameters as listed in the table below:

<table>
<thead>
<tr>
<th>Example</th>
<th>( R )</th>
<th>( \lambda_{\text{cn}} )</th>
<th>( \lambda_{\text{all}} )</th>
<th>( q )</th>
<th>( Q_{\text{max}} )</th>
<th>( T )</th>
<th>( K )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example A</td>
<td>4</td>
<td>0.8</td>
<td>1.00</td>
<td>4</td>
<td>20</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>Example B</td>
<td>3</td>
<td>0.4</td>
<td>0.70</td>
<td>4</td>
<td>20</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Example C</td>
<td>2</td>
<td>0.777777</td>
<td>0.50</td>
<td>8</td>
<td>20</td>
<td>12</td>
<td>1</td>
</tr>
</tbody>
</table>

The parameter \( \lambda_{\text{all}} \) in this table is the traffic generated by all \( R + 1 \) sources, all other parameters are as defined before.

\[
\lambda_{\text{all}} = (R + 1) \frac{\beta}{\alpha + \beta} \lambda_{\text{cn}}. \tag{55}
\]

To obtain the analytical results, the modified algorithm (see below) was used. All simulation results are obtained by running ten independent runs, in each of which 10,000,000 packets are generated. Confidence intervals are calculated with a confidence coefficient of 0.95. The obtained packet discard and cell loss ratios are shown in the table below. The results obtained by using the analytical model are of the same order as the results obtained by simulation.
In the algorithm as proposed in Section 4 we calculate in every iteration step the stationary distributions $\psi$ and $\phi$ (step 3) and $\tilde{\psi}$ and $\tilde{\phi}$ (step 4). Steps 2–4 are repeated several times because of step 5. From those steps, step 4 is clearly the most time consuming one. As we have noticed in several examples, our results will be of the same order if we change step 2(b) in

\[2(b):\text{ Calculate all the values } p_i(u, v, j, k, l) \text{ and } q_d(u, v, i, j, k, l) \text{ using } \phi \text{ and } \psi, \]

skip step 4(b) and only use $\tilde{\phi}$ and $\tilde{\psi}$ at the end of the algorithm (step 6). This however makes a significant difference in computation time. To illustrate that the results of the modified algorithm are of the same order as the results of the original algorithm, consider the plots of Figure 2. This figure shows the distribution of the total queue length and of a VQ queue length for three examples with the following parameters:

<table>
<thead>
<tr>
<th>$R$</th>
<th>$\lambda_{in}$</th>
<th>$\lambda_{out}$</th>
<th>$q$</th>
<th>$Q_{max}$</th>
<th>$T_h$</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example D</td>
<td>2</td>
<td>0.5</td>
<td>0.75</td>
<td>5.5</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>Example E</td>
<td>2</td>
<td>0.77777</td>
<td>0.39</td>
<td>8.0</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>Example F</td>
<td>3</td>
<td>1.0</td>
<td>1.00</td>
<td>5.0</td>
<td>20</td>
<td>8</td>
</tr>
</tbody>
</table>

The obtained packet discard and cell loss ratios are summarized in the table below:

<table>
<thead>
<tr>
<th>$P_{\text{disc}}$</th>
<th>CLR</th>
<th>$P_{\text{disc}}$</th>
<th>CLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original algorithm</td>
<td>Modified algorithm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example D</td>
<td>1.9594e-02</td>
<td>1.4075e-04</td>
<td>2.1756e-02</td>
</tr>
<tr>
<td>Example E</td>
<td>3.5613e-03</td>
<td>2.6224e-03</td>
<td>3.6210e-03</td>
</tr>
<tr>
<td>Example F</td>
<td>5.8654e-02</td>
<td>2.0118e-02</td>
<td>6.1643e-02</td>
</tr>
</tbody>
</table>

Since we are more interested in the evolution of the curves if parameters are changed than in the absolute values, all following numerical results presented are obtained with the modified algorithm. Figure 3 shows the impact on the queue length distributions if the same total amount ($\lambda_{all} = 0.7$) of traffic is generated by a different number of sources $R + 1$. Figure 4 shows the impact on the packet discard ratio and the cell loss ratio. If there are more sources, the burstiness of the generated traffic is larger, which enlarges the probability that $Q$ exceeds the EPD threshold $Th$ (see Figure 3). If there are more sources, the probability that $N$, the number of active VCs, will be larger increases. But a larger $N$ means also a smaller threshold $Th$, so more packets will be discarded if there are more sources. However, the larger $N$, the smaller the effect on $Th$ if $N$ evolves from $N$ to $N + 1$. This explains the smaller increase of $P_{disc}$ for growing $N$ in Figure 4. Although more packets are discarded if the number of sources increases, the cell loss ratio does not decrease, but on the contrary increases. This is because a larger burstiness of the traffic has a larger negative effect on the cell loss than that a small reduction of the amount of traffic that enters the buffer has a positive effect.

Figures 5 and 6 show the impact of the EPD threshold $Th$ on the queue distributions, the packet discard ratio and the cell loss ratio for a scenario with offered load $\lambda_{all} = 0.7$. As can be seen from Figure 5, the influence of a change of $Th$ on the queue length distribution of one VC is only a minor one. We noticed from results with a larger offered load $\lambda_{all}$ that this influence becomes larger if $\lambda_{all}$ increases. As can be seen in Figure 5, the curves of the total queue length distribution start to diverge downwards from the other curves from the value $Th$ on, since once the total queue is over this threshold, it becomes possible to discard packets, so less of the generated traffic will effectively enter the buffer. As expected, an increase of $Th$ leads to a decrease of $P_{disc}$, as shown in Figure 6. This is however at the cost of an increasing cell loss ratio, since a larger part of the traffic is allowed to enter the buffer.

7 Conclusions

This report adresses the EPD with per-VC queueing mechanism. This mechanism was proposed in [10] because the EPD scheme with a single FIFO queue can not guarantee fairness between
competing flows. In this report we developed an analytical approximation model for the mechanism. It combines the source models used in the literature for EPD and PPD and the approximate vacation analysis often used for cyclic service systems.

To validate our model, the obtained results were compared with simulation results. Both methods lead to results that are shown to have the same order of magnitude. So they can be used very well to study the evolution of performance measures such as buffer occupation, packet discard ratio and cell loss ratio when parameters of the input traffic or of the discarding mechanism are changed. In this report we looked at the effect of a change in the number of sources which generate the same amount of traffic and of a change in the setting of the EPD threshold $Th$. In the first scenario we saw that the important influence of increasing burstiness on the CLR could not be overthrown by the increasing packet discard ratio of the discarding scheme. The second scenario indicated the minor influence of $Th$ on the queue occupation of a VC queue, and the expected influence (decreasing packet discard ratio and increasing cell loss ratio for increasing $Th$) on packet discard ratio and cell loss ratio.

Future work will include the investigation of the influence of other parameters on the performance, and a comparison between the results obtained by this packet discard scheme and the EPD scheme.

Figure 3: Queue length distributions when the same amount of traffic ($\lambda_{all} = 0.7$) is generated by a different number of sources $R + 1$.

Figure 4: Evolution of $P_{disc}$ and CLR when the same amount of traffic ($\lambda_{all} = 0.7$) is generated by a different number of sources $R + 1$. 
Figure 5: Queue length distributions for different values of the EPD threshold $Th$.

with a single FIFO used in [7].

References


Figure 6: Evolution of $P_{\text{disc}}$ and CLR for different values of the EPD threshold $Th$. 


