The Ballot Theorem Strikes Again: Packet Loss Process Distribution

Omer Gurewitz, Moshe Sidi, *Senior Member, IEEE*, and Israel Cidon, *Senior Member, IEEE*

***Abstract*—The probability distribution of the number of lost packets within a block of consecutive packet arrivals into a finite buffer is an important quantity in various networking problems. In a recent paper, Cidon, Khamisy, and Sidi introduced a recursive scheme to derive this distribution. In this paper, we derive explicit expressions for this distribution using various versions of the powerful *Ballot Theorem*. The expressions are derived for a single source M/M/1/K queue.**

***Index Terms*—Ballot theorem, blocking probability, finite queues, forward error recovery, high-speed networks, packet loss processes.**

I. Introduction

In many networking applications, messages are divided into smaller data units—packets—that are transmitted consecutively into the network. The packet loss process is an important performance measure in the design and analysis of such applications. The loss of a single packet of the message may result in the loss of the entire message. Forward error-correcting techniques are suggested that use block encoding and hence the loss probability of j packets out of a message (block) of n consecutive packets, is a crucial measure for proper design of error recovery mechanisms [1].

In a recent paper, Cidon, Khamisy, and Sidi [1] introduced recursive schemes to compute the above quantity for various packet-arrival models and service distributions. In their most basic model, the arrival process is Poisson and service times are exponentially distributed, resulting in the recursive analysis of the above probabilities for a finite M/M/1 queue. Later, Altman and Jean-Marie [2] used the recursive equations to derive expressions for the multidimensional generating function of the required probabilities and for some system parameters obtained explicit expressions for the probabilities themselves. The method they introduced is algebraic in nature.

In this paper, we introduce an alternate method (that also yields alternate simpler expressions) to obtain explicit expressions for the required probabilities for all the system parameters of interest. The method is probabilistic in nature and is based on extensive use of various versions of the powerful *Ballot Theorem* [3]. There are several other examples in which Ballot Theorems have been used to analyze the performance of networking systems [4], [5].

Manuscript received May 14, 1999; revised May 28, 2000. This work was supported by the Consortium for Broadband Communication administered by the Chief Scientist of the Israeli Ministry of Commerce and Industry.

The authors are with the Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: gurewitz@tx.technion.ac.il; moshe@ee.technion.ac.il; cidon@ee.technion.ac.il).

Communicated by V. Anantharan, Associate Editor for Communication Networks.

Publisher Item Identifier S 0018-9448(00)09682-6.

II. The Model and Preliminaries

We consider systems with variable-length packets whose transmission time is exponentially distributed with parameter μ. The packets are stored in a queue that can accommodate up to m packets (including the served packet) and are served (transmitted) according to the first-in-first-out (FIFO) rule. If a packet arrives at a system that contains m packets, it is lost. The packets are grouped into fixed size blocks, namely, every n consecutive packets form a *block* and we are interested in the probability distribution of the number of lost packets within a block in steady state. Other grouping scenarios were also considered in [1].

Packets arrive at the system according to a Poisson process with rate λ. The average load ρ is defined as $\rho \triangleq \lambda/\mu$. We recall that the stationary probability of having i packets in the system at an arrival epoch (and also at an arbitrary epoch) is

$$\Pi(i) = \rho^i / \sum_{l=0}^{m} \rho^l, \qquad 0 \le i \le m.$$

Our purpose in this paper is to compute the probabilities $P_i(j,n), i = 0,1,\ldots,m, n \ge 1, 0 \le j \le n,$ of j losses in a block of n packets, given that there are i packets in the system just before the arrival epoch of the first packet in the block, as well as the probabilities $P(j,n), n \ge 1, 0 \le j \le n,$ of j losses in a block of n consecutive packets. Since the first packet in a block is arbitrary,[1] , we have the relation

$$P(j,n) = \sum_{i=0}^{m} \Pi(i) P_i(j,n). \tag{1}$$

In [1], a recursive method has been presented to compute the probabilities $P_i(j,n), i = 0,1,\ldots,m, n \ge 1, 0 \le j \le n$ and (1) was used to obtain $P(j,n)$. In this paper, we introduce a probabilistic method to obtain all these probabilities explicitly.

Obviously, when j packets are lost $(j \ge 1)$, one of the n packets of the block is the jth packet to be lost, and no other packet of the block is lost after that packet. Let $P_i^k(j,n)$ denote the probability that the kth packet of the block is the jth and the last packet of that block to be lost, given that there are i packets

[1]Note that the block structure is an indexing imposed on the sequence of (Poisson) arrivals starting at an arbitrary location, hence, the M/M/1 behavior is intact.

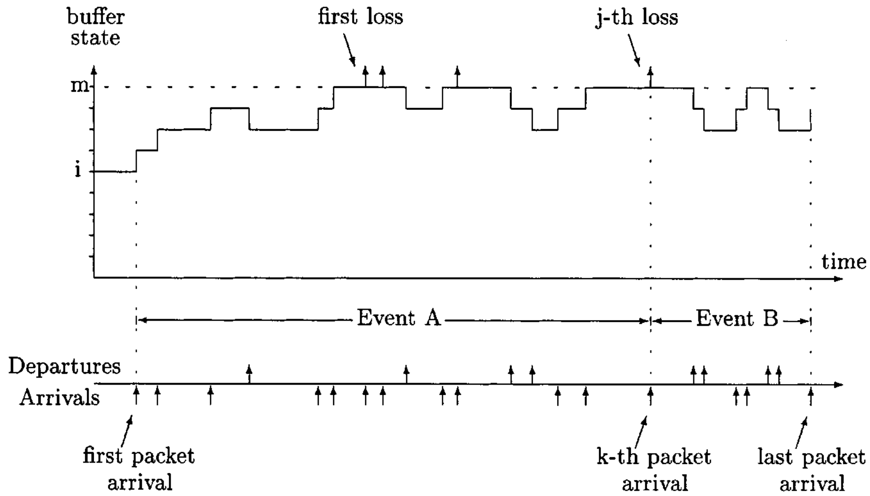

Fig. 1. An example to illustrate Events A and B.

in the system just before the arrival epoch of the first packet of the block. It is clear that

$$P_i(j,n) = \sum_{k=1}^{n} P_i^k(j,n) \tag{2}$$

Note that

$$\sum_{k=1}^{n} P_i^k(j,n) = \sum_{k=j}^{n} P_i^k(j,n)$$

since the probability that the kth packet of the block is the jth to be lost, for $k < j$ is zero.

For the kth packet in the block to be the jth and the last packet of the block to be lost, two mutually exclusive events must occur (see Fig. 1).

Event A: The kth packet in the block is the jth lost packet given that upon the arrival of the first packet of the block the buffer contains i packets.

Event B: No more packets of the block are lost after the kth packet is lost.

In order to proceed we define a *path* as an ordered sequence of events. Event is either an arrival $\{a\}$ or a departure $\{d\}$. For instance, the ordered sequences $\{aada\}$ and $\{aaad\}$ are both paths that contain three arrivals and one departure. They are different paths since the order of events is different.

Let p denote the probability that an arrival precedes a departure when the queue is not empty and $q = 1 - p$. Due to the exponential nature of the service times and the interarrival times, we have that

$$p = \frac{\lambda}{\lambda + \mu} = \frac{\rho}{1+\rho}$$
$$q = \frac{\mu}{\lambda + \mu} = \frac{1}{1+\rho}.$$

The memoryless property of the exponential distribution also implies that at any epoch the probability that a path will have u arrivals and v departures is $p^u q^v$ if along this path the buffer never empties. If the buffer empties h times $(h \le u)$ along this path, then the probability of such a path increases to $p^{u-h}q^v$ (since when the buffer is empty, the next event is an arrival with probability one).

In the sequel we will compute the probabilities of Events A and B. To that end we will need the following Lemmas.

Lemma 1: Assume that upon the arrival of the first packet of the block the buffer contains i packets. The kth packet is the jth to be lost if and only if upon the arrival of the kth packet the number of arrivals exceeds the number of departures by $m-i+j$ for the *first* time.

Proof: Assume that the kth packet is the jth to be lost. This implies that the buffer becomes full at least j times right after the arrival of packets $1, 2, \ldots, k$. Let $r_0 (r_0 < k)$ be the packet that fills the buffer for the first time. Clearly, no packets from the block are lost prior to and including the arrival of r_0, so they all join the buffer. Right before the arrival of the first packet there are $m - i$ empty places in the buffer, so for r_0 to fill it for the first time, the number of arrivals must exceed the number of departures by exactly $m - i$ (counting from the arrival of the first packet of the block until the arrival of r_0).

Let r_1 be the first packet of the block to be lost. Counting events from the time the buffer was full (right after the arrival of packet r_0) until r_1 is lost, it is clear that during this period the number of arrivals exceeds the number of departures by one. This is true since just before the arrival of r_1 the buffer was full, i.e., the total content of the buffer did not change, no packet was lost, so the number of arrivals equals the number of departures. Using this reasoning inductively, we conclude that upon each packet that is lost, the number of arrivals exceeds the number of departures by one counting from the previous loss (since a lost packet leaves the buffer full). Since there are exactly j losses upon the arrival of the kth packet we conclude that the number of arrivals exceeds the number of departures by $m - i + j$ for the first time upon the arrival of the kth packet, completing the proof of one direction.

For the other direction, consider the first time the number of arrivals exceeds the number of departures by $m - i + j$. This can happen only upon an arrival, say of the kth packet. We need to show that this arrival is lost and it is the jth loss. To see that consider first the case $j = 1$ and let u be the packet that causes the number of arrivals to exceed the number of departures by $m - i + 1$ for the first time. This implies that prior to the arrival of u the number of arrivals never exceeded the number of departures by more than $m - i$. Since upon the arrival of the first packet there are $m - i$ spaces in the buffer, no packets were lost prior to the arrival of u. In addition, just before the arrival of u the number of arrivals exceeds the number of departures by $m - i$, implying the buffer is full, so packet u is lost and is the first to be lost. Observing that each lost packet leaves the buffer full, we use the above reasoning, inductively, to conclude that if we look at the first time the number of arrivals exceeds the number of departures by $m - i + j$, then a packet is lost and it is the jth loss, completing the proof of the other direction. □

Lemma 2: Assume the kth packet is lost. No packets are lost afterwards if and only if throughout the arrival of the rest of the block the number of departures is always not smaller than the number of arrivals (counting starts right after the arrival of the kth packet).

Proof: Assume that the number of departures is never behind the number of arrivals during the arrival of the rest of the block and packet $l > k$ is the first to be lost after packet k. Starting counting when the buffer is full just after packet k is lost, Lemma 1 dictates that for packet l to be lost upon its arrival, the number of arrivals should exceed the number of departures by one. However, this contradicts the fact that the number of departures is never behind the number of arrivals during the arrival of the rest of the block, proving that no packets from the rest of block can be lost, completing the proof of one direction.

For the other direction assume that no packets from the rest of the block are lost and the number of arrivals does exceed the number of departures for the first time upon the arrival of packet $l > k$. Using Lemma 1, start counting when the buffer is full just after packet k is lost, if upon an arrival of a packet the number of arrivals exceeds the number of departures, the packet is lost. So packet l must be lost, but this contradicts the fact that no packets from the rest of the block are lost, thus completing the proof of the other direction. □

III. Analysis

In the analysis we distinguish between the simple case in which the buffer can contain a complete block $(m \geq n)$ and the more complicated case that the block is larger than the buffer $(n > m)$.

A. CASE i): The Buffer Can Contain a Complete Block $(m \geq n)$

The case that the buffer can contain a complete block $(m \geq n)$ is relatively simple, since if the buffer ever becomes empty during the arrival of the packets of the block, no further packets from the currently arriving block will be lost, since there is enough room for all packets. Therefore, if the kth packet of the block is lost, the buffer could not have been empty at any of the arrival epochs of the preceding packets of the block.

In the following we start with the computations of the probabilities $P_i(j, n)$ that together with (1) yield $P(j, n)$. We then provide a much simpler expression for $P(j, n)$ directly.

1) Computing $P_i(j, n)$: We start by computing the probability of Event A, i.e., the probability that the kth packet in the block is the jth lost packet. From Lemma 1, it follows that we are interested in all the paths that end with an arrival and contain k arrivals $(k \leq n)$ and l departures such that $k - l = m - i + j$ (i is the number of packets in the buffer upon the arrival of the first packet of the block). Such paths will never cause the buffer to become empty, irrespective of the order of events of that path since

$$l = k - (m - i + j) \leq n - (m - i + j) \leq i - j < i$$

namely, the number of departures is smaller than the initial content of the buffer. Furthermore, exactly j packets are lost along such a path. All these paths have $2k - g (g \stackrel{\Delta}{=} m - i + j)$ events of which the first is an arrival (of the first packet of the block) and they are all equiprobable. The probability of each of these paths is $p^{k-1} q^{k-g}$ (first arrival with probability one, $k - 1$ more arrivals each with probability p and $k - g$ departures each with probability q) and their total number is $\binom{2k-g-1}{k-1}$. Recall, however, that due to Lemma 1 we need to count only those paths for which the kth arrival is the jth loss, i.e., the paths in which the number of arrivals exceeds the number of departures by g for the *first* time in the kth arrival epoch. To count these paths we use the following version of the Ballot Theorem [3].

Classial Ballot Theorem: In a ballot candidate A scores α votes and candidate B scores β votes, where $\alpha > \beta$. Assuming that all orderings are equally likely, the probability that throughout the counting A is always ahead in the count of votes is $(\alpha - \beta)/(\alpha + \beta)$.

To count the relevant paths, the theorem is used looking from the kth arrival epoch backward. In this backward direction, the number of arrivals must always exceed the number of departures (otherwise, at some point these two numbers were equal, contradicting the "first-time" requirement when looking at the path in the forward direction). From the Ballot Theorem we have that the proportion of the number of paths with $k - 1$ arrivals and $k - g$ departures such that the number of arrivals must always exceed the number of departures (in the backward direction) out of the total number of paths with $k - 1$ arrivals and $k - g$ departures is $(g - 1)/(2k - g - 1)$. Consequently, the probability that upon k arrivals and $k - g$ departures, the number of arrivals exceeds the number of departures by g for the first time in the kth arrival epoch (Event A) is

$$\frac{g-1}{2k-g-1} \binom{2k-g-1}{k-1} p^{k-1} q^{k-g}. \tag{3}$$

It now remains to compute the probability of Event B, i.e., the probability that no more packets of the block are lost after the kth packet is lost. Since upon the arrival of the kth packet the buffer is full (otherwise, the kth packet would not have been lost), we conclude from Lemma 2 that we should consider only those paths in which the number of departures is not smaller than the number of arrivals until the last packet of the block arrives. Furthermore, once the number of departures is $n - k$, the rest of the path is irrelevant since enough space is available to insure that no more packets will be lost. Therefore, we consider only those paths for which the number of departures is $n-k$ and the number of arrivals can obtain any value between 0 and $n-k$. Note that the buffer does not become empty for such paths, since it starts with a full buffer and the number of departures is smaller than the buffer size $m(n-k<n\leq m)$. To insure we count disjoint paths, we consider paths for which the last event is a departure. The probability of each path with $n-k$ departures, l arrivals, and the last event is a departure is $qp^lq^{n-k-1}=p^lq^{n-k}$ and their number is $\binom{n-k-1+l}{l}$. Note that $0\leq l\leq n-k-1$ since the last event is a departure. The proportion of the number of paths in which the number of departures is not smaller than the number of arrivals out of the total number of paths with $n-k$ departures and l arrivals is again obtained using the Ballot Theorem to yield $(n-k-l)/(n-k)$. Combining the above we obtain the probability of Event B, i.e., the probability that no more packets of the block are lost after the kth packet is lost

$$\sum_{l=0}^{n-k-1}\frac{n-k-l}{n-k}\binom{n-k-1+l}{l}p^lq^{n-k}. \tag{4}$$

Combining (2)–(4) and using the fact that $p_i^k(j,n)=0$ for $1\leq k<m-i+j$ (since $m-i$ packets of the block can be accommodated into the buffer and cannot be lost) we obtain for $i=0,1,\ldots,m,\ n\geq 1, 1\leq j\leq n$

$$P_i(j,n)=\sum_{k=g}^{n}\frac{g-1}{2k-g-1}\binom{2k-g-1}{k-1}p^{k-1}q^{k-g}\times\sum_{l=0}^{n-k-1}\frac{n-k-l}{n-k}\binom{n-k-1+l}{l}p^lq^{n-k}$$

where we recall that $g=m-i+j$, and we define an empty sum to be one and for $k=g=1$ define the first term as one as well ($g=1$ if and only if $i=m$ and $j=1$, i.e., when the first packet is lost). Obviously,

$$P_i(0,n)=1-\sum_{j=1}^{n}P_i(j,n).$$

From the above discussion it is clear that if we are interested in the probability of losing *at least* j packets out of a block of n packets given that there were i packets in the buffer upon the arrival of the first packet of the block (let $P_i(\geq j,n)$ denote this probability), then we only have to sum (3) over all k, since Event B is irrelevant in this case. The result is

$$P_i(\geq j,n)=\sum_{k=g}^{n}\frac{g-1}{2k-g-1}\binom{2k-g-1}{k-1}p^{k-1}q^{k-g}$$

where for $k=g=1$ the first term is defined as one. Obviously,

$$P_i(j,n)=P_i(\geq j,n)-P_i(\geq j-1,n)$$

so we have an alternate expression to compute $P_i(j,n)$ that requires only $O(n-j+1)$ computations. The fact that our expression requires less computations as j increases is not surprising, since we are looking at the last packet to be lost, and as j increases, the number of ways to choose that packet decreases.

2) Computing $P(j,n)$: Using a similar approach to that presented in the previous section we can compute the probabilities $P(j,n)$ directly. In fact, we can show that the probabilities of losing at least j packets out of a block of n packets, $P(\geq j,n)$, is given by

$$\begin{aligned}P(\geq j,n)=&\ \Pi(m)\sum_{h=j}^{n}\frac{j-1}{2h-j-1}\binom{2h-j-1}{h-1}\\&\times p^{h-1}q^{h-j}(n-h+1)\\&-\Pi(m)\sum_{h=j+1}^{n}\frac{j}{2h-j-2}\binom{2h-j-2}{h-1}\\&\times p^{h-1}q^{h-j-1}(n-h+1).\end{aligned} \tag{5}$$

Obviously,

$$P(j,n)=P(\geq j,n)-P(\geq j-1,n)$$

so we have a very simple expression to compute $P(j,n)$ that requires only $O(m+n-j+1)$ computations (the m comes from the need to compute $\Pi(m)$) which is a substantial improvement over of the complexity of $O(m+nj^2)$ computations required by the expressions obtained in [2].

To derive (5) we let $X^j_{l_1,l_h}$ ($l_1,l_2,\ldots,l_h$ are consecutive packets in the block) be the set of all paths in which packets l_1 and $l_h (j\leq h\leq n)$ are lost and $j-2$ packets are lost between them (so a total of j packets are lost). Each of these paths can be divided into two parts. a) The loss of packet l_1 whose probability is $\Pi(m)$. b) The rest of the path that starts right after the loss of packet l_1 and ends with the loss of packet l_h which is the jth loss. The probability of this part is the probability of a path that contains $h-1$ arrivals (packets $l_2,l_3,\ldots,l_h$), $h-j$ departures, and the first time the number of arrivals exceeds the number of departures by $j-1$ (remember l_1 was lost, so we need another $j-1$ losses) is after $2h-j-1$ events. This probability has been derived in (3) and is given by

$$(j-1)/(2h-j-1)\binom{2h-j-1}{h-1}p^{h-1}q^{h-j}.$$

Therefore,

$$\mathrm{Prob}[X^j_{l_1,l_h}]=\Pi(m)\frac{j-1}{2h-j-1}\binom{2h-j-1}{h-1}p^{h-1}q^{h-j}.$$

Note that the above probability does not depend on l_1 and l_h but only on the number of packets h between them. The number of possible pairs of packets that are h places apart within a block of size n is $n-h+1$. However, if we sum the above probability over all possible pairs $\sum_{h=j}^{n}\mathrm{Prob}[X^j_{l_1,l_h}](n-h+1)$ a path that contains $j+t$ $(0\leq t\leq n-j)$ losses will be counted $t+1$ times (once for every j consecutive losses). Therefore, in order to insure that each path is counted exactly once, we need to subtract each path that contains $j+t$ losses t times. Using the same reasoning if we compute $\sum_{h=j+1}^{n}\mathrm{Prob}[X^{j+1}_{l_1,l_h}](n-h+1)$, each path that contains $(j+1)+t$ losses will be counted $t+1$ times, but this is exactly what we need to subtract, hence (5).

Remark: It is easy to see that the expressions derived above can be used as long as $n\leq m+j$ since, for these parameters, if the buffer ever becomes empty during the arrival of the packets of the block, there is no possibility that the number of arrivals will exceed the number of departures by $m+j$.

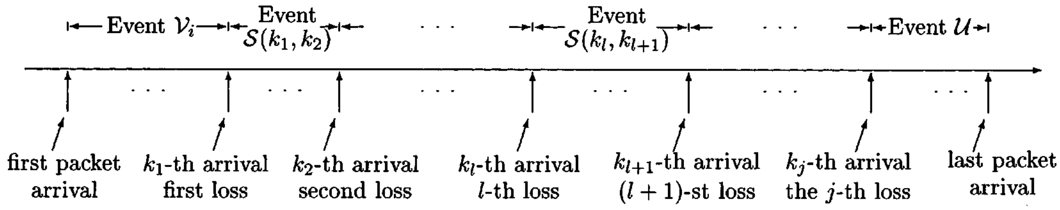

Fig. 2. A typical path with j losses.

B. *CASE ii): The Block is Larger than the Buffer* $(n > m)$

The analysis presented in this section is rather general and holds for any set of parameters n, m. Yet, it is more complex than the analysis presented for Case i) above, and the expressions obtained are much more complex. Note that Altman and Jean-Marie [2] did not provide expressions for $P(j, n)$ for blocks larger than the buffer. In this section, we also derive $P_i(j, n)$ first and then show the derivation of $P(j, n)$.

As we observed from Lemma 1, if upon the arrival of the first packet of the block the buffer contains i packets, to lose $j (j \geq 1)$ packets we must have that the number of arrivals exceeds the number of departures by $m - i + j$ (counting from the arrival of the first packet of the block until the arrival of the jth lost packet). The main difficulty in case the block is larger than the buffer is that paths with this property may cause the buffer to become empty (even several times). As explained in Section II, the probability of a path depends on the number of times the buffer empties along this path, and counting the number of paths is rather complex.

To carry the analysis consider a typical path (see Fig. 2) that starts upon an arrival of the first packet of the block when there are i packets in the buffer, contains n packet arrivals of which j $(j \geq 1)$ are lost. Denote the lth lost packet by $k_l (1 \leq l \leq j,$ $k_1 < k_2 < \cdots < k_j)$. We note that such a path can be decomposed into three types of mutually exclusive events as follows.

Event $\mathcal{V}_i(k_1)$: the first packet to be lost is k_1 given that upon an arrival of the first packet of the block there are i packets in the buffer.

Event $\mathcal{S}(k_l, k_{l+1})$: packet k_{l+1} is lost given that packet k_l was lost.

Event $\mathcal{U}(k_j)$: packet k_j is the last to be lost.

Clearly, a path is a succession of events

$$\mathcal{V}_i(k_1), \mathcal{S}(k_1, k_2), \mathcal{S}(k_2, k_3), \ldots, \mathcal{S}(k_{j-1}, k_j), \mathcal{U}(k_j).$$

Let $v_i(k_1), s(k_l, k_{l+1})$ and $u(k_j)$ be the probabilities of events $\mathcal{V}_i(k_1), \mathcal{S}(k_l, k_{l+1})$ and $\mathcal{U}(k_j)$, respectively. It is clear that

$$P_i(1, n) = \sum_{k_1=1}^{n} v_i(k_1) u(k_1)$$

$$P_i(2, n) = \sum_{k_1=1}^{n-1} \sum_{k_2=k_1+1}^{n} v_i(k_1) s(k_1, k_2) u(k_2)$$

and in the general case

$$P_i(j, n) = \sum_{k_1=1}^{n-j} \sum_{k_2=k_1+1}^{n-j+1} \cdots \sum_{k_j=k_{j-1}+1}^{n} v_i(k_1) s(k_1, k_2) \ldots s(k_{j-1}, k_j) u(k_j). \quad (6)$$

By definition, $s(k_l, k_{l+1}) = 0$ when $k_l \geq k_{l+1}$ for $1 \leq l \leq j-1$. Therefore, (6) can be rewritten as

$$P_i(j, n) = \sum_{k_1=1}^{n} \sum_{k_2=1}^{n} \ldots \sum_{k_j=1}^{n} v_i(k_1) s(k_1, k_2) \ldots s(k_{j-1}, k_j) u(k_j)$$

or in a matrix form

$$P_i(j, n) = \boldsymbol{V}_i \cdot \boldsymbol{S}^{j-1} \cdot \boldsymbol{U}^T \quad (7)$$

where $\boldsymbol{V}_i$ and $\boldsymbol{U}$ are n-length row vectors with elements $v_i(\cdot)$ and $u(\cdot)$, respectively, and $\boldsymbol{S}$ is an $n \times n$ matrix with elements $s(\cdot, \cdot)$. Note that the matrix $\boldsymbol{S}$ is very simple in the sense that each of its rows (except the first) is identical to the previous row with the elements moved one place to the right. Therefore, the computation of $\boldsymbol{S}^j$ needs only $O(j \cdot n^2)$ operations.

We now have to compute the elements of $\boldsymbol{V}_i$, $\boldsymbol{U}$, and $\boldsymbol{S}$. To that end we consider a general path that starts when there are $\iota (1 \leq \iota \leq m)$ packets in the buffer, ends with $\mu (1 \leq \mu \leq m)$ packets in the buffer, contains $\eta (0 \leq \eta \leq 2n + \iota - \mu)$ events (arrivals and departures), and no packets are lost. Let $\zeta_\eta(\iota, \mu)$ be the probability of such a path. We can express the values of the elements of $\boldsymbol{V}_i, \boldsymbol{U}$ and $\boldsymbol{S}$ using the quantities $\zeta_\eta(\iota, \mu)$ as follows.

Expression for $v_i(k_1)$:

If the first packet of the block arrives at a full buffer, it is lost and no other packet can be the first to be lost. Otherwise, that first packet joins the buffer so the evolution of the path starts with $\iota = i + 1$ and $n - 1$ packets still to arrive. For the k_1 packet to be the first lost packet, there must be an epoch after the arrival of the $k_1 - 1$ packet that the buffer is full, i.e., $\mu = m$ and the next event is an arrival. Up to that epoch, the number of arrivals must exceed the number of departures by exactly $m - i - 1$ (otherwise, the buffer would not be full). The number of arrivals up to that epoch is $k_1 - 2$ (from the second arrival until the arrivals of packet $k_1 - 1$), so the number of departures is $k_1 - m + i - 1$ to yield a total number of $\eta = 2k_1 - m + i - 3$ events. Combining the above we get (8) at the bottom of the following page.

Expression for $s(k_l, k_{l+1})$:

Right after the arrival and loss of packet k_l the buffer is full so $\iota = m$. Right before the arrival of packet k_{l+1} the buffer must be full, i.e., $\mu = m$ and the next event is an arrival (of packet k_{l+1}) so we have to multiply our probability by p. Since we start and end with a full buffer, the number of arrivals $k_{l+1} - k_l - 1$ equals the number of departures and the total number of events is $\eta = 2(k_{l+1} - k_l - 1)$. Therefore,

$$s(k_l, k_{l+1}) = p \cdot \zeta_{2(k_{l+1} - k_l - 1)}(m, m) \quad (9)$$

Expression for $u(k_j)$:

It is clear that if k_j is the last packet of the block $(k_j = n)$ then no packets can be lost after it, i.e., $u(n) = 1$. We now consider the cases for which $k_j < n$. From the proof of Lemma 2 we conclude that in order to compute $u(k_j)$ it suffices to consider the path until the epoch that the $n-k_j$th departure occurs, starting to count after the arrival of packet k_j (since after that there is sufficient buffer space for the rest of the block). In order to avoid counting a path more than once, we consider only those paths that end with a departure and look at the paths right before that departure, so our probability should be multiplied by q to take care of that departure. So the total number of departures we consider is $n-k_j-1$. Let h be the total number of arrivals $(h \leq n-k_j-1)$ to avoid losses and $h \geq \max(0, n-k_j-m)$ since the difference between the number of departures, including the last one, and the number of arrivals cannot exceed m, so the total number of events is $\eta = n-k_j-1+h$ and we have

$$\mu = m-(n-k_j-1)+h = m-n+k_j+h+1$$

packets in the buffer. Consequently, we get (10) at the bottom of this page

To complete the analysis we need to compute $\zeta_\eta(\iota, \mu)$. The main difficulty in computing these probabilities is that for a given number of arrivals and a given number of departures, the number of times the buffer may become empty depends on η, ι and μ.

We begin by considering paths that do not cause the buffer to ever become empty. Let $\xi_\eta(\iota, \mu)$ be the probability of such paths $(1 \leq \iota \leq m, 1 \leq \mu \leq m, |\iota-\mu| \leq \eta \leq 2n+\iota-\mu$ and by definition $\xi_0(\iota,\mu)=1$ for $\iota=\mu$ and $\xi_0(\iota,\mu)=0$ for $\iota \neq \mu)$. Such a path contains $(\eta-\iota+\mu)/2$ arrivals and $(\eta+\iota-\mu)/2$ departures. Therefore,

$$\xi_\eta(\iota,\mu) = \mathcal{N}_\eta(\iota,\mu) p^{\frac{\eta-\iota+\mu}{2}} q^{\frac{\eta+\iota-\mu}{2}} \tag{11}$$

where $\mathcal{N}_\eta(\iota,\mu)$ is the number of such paths. This number can be computed by applying the following Ballot problem [3].

Ballot Theorem*:* In a ballot, candidate A scores a votes and candidate B scores b votes, and all the possible voting records are equally probable. Let $c-d < a-b < c$ where $0 < c < d$ are integers. The number of possible arrangements of votes such that candidate A is always ahead of candidate B by less than c votes and more than $c-d$ votes is

$$\sum_{\Upsilon} \left[\binom{a+b}{b-\Upsilon d} - \binom{a+b}{b+c-\Upsilon d} \right]$$

where $-\infty < \Upsilon < \infty$ takes values in the above sums so that the binomial coefficients are proper (e.g., in the first sum $\Upsilon d < b$ and $a > -\Upsilon d$).

The quantity $\mathcal{N}_\eta(\iota,\mu)$ can now be computed by identifying a with arrivals, b with departures, c with the threshold by which the number of arrivals must exceed the number of departures in order to lose a packet, and $c-d$ with the threshold by which the number of arrivals must be behind the number of departures in order to empty the buffer, i.e., $a=(\eta-\iota+\mu)/2, b=(\eta+\iota-\mu)/2$, $c=m+1-\iota$, and $d=m+1$ to obtain

$$\mathcal{N}_\eta(\iota,\mu) = \sum_{\Upsilon} \left[\binom{\eta}{\frac{\eta+\iota-\mu}{2} - \Upsilon(m+1)} - \binom{\eta}{\frac{\eta-\iota-\mu}{2} - \Upsilon(m+1)} \right]$$

or using (11)

$$\xi_\eta(\iota,\mu) = \sum_{\Upsilon} \left[\binom{\eta}{\frac{\eta+\iota-\mu}{2} - \Upsilon(m+1)} - \binom{\eta}{\frac{\eta-\iota-\mu}{2} - \Upsilon(m+1)} \right] p^{\frac{\eta-\iota+\mu}{2}} q^{\frac{\eta+\iota-\mu}{2}} \qquad \iota \geq 1, \quad \mu \geq 1. \tag{12}$$

Using (12) we can easily compute the probability of paths with η events that start with an empty buffer $(\iota = 0)$, end with $\mu \geq 1$ packets in the buffer, and do not become empty in between. Let $\xi_\eta(0,\mu)$ be this probability. Then since the next event when the buffer is empty is an arrival with probability one we have

$$\xi_\eta(0,\mu) = \xi_{\eta-1}(1,\mu), \qquad \mu \geq 1. \tag{13}$$

Similarly, we can compute the probability of paths with η events that start with $\iota \geq 1$ packets in the buffer, end with an empty buffer $(\mu = 0)$ and do not become empty in between. Let $\xi_\eta(\iota, 0)$ be this probability. Then since the buffer becomes empty upon a departure that occurs with probability q we have

$$\xi_\eta(\iota,0) = q\xi_{\eta-1}(\iota,1), \qquad \iota \geq 1. \tag{14}$$

Combining (13) and (14) we obtain the probability of paths that start and end with an empty buffer and do not become empty in between

$$\xi_\eta(0,0) = q\xi_{\eta-2}(1,1). \tag{15}$$

$$v_m(k_1) = \begin{cases} 1, & k_1 = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$v_i(k_1) = \begin{cases} 0, & k_1 \leq m-i \\ p \cdot \zeta_{2k_1-m+i-3}(i+1,m), & \text{otherwise} \end{cases} \quad i \neq m. \tag{8}$$

$$u(k_j) = \begin{cases} q \displaystyle\sum_{h=\max(0,n-k_j-m)}^{n-k_j-1} \zeta_{n-k_j-1+h}(m, m-n+k_j+h+1), & k_j < n \\ 1, & k_j = n. \end{cases} \tag{10}$$

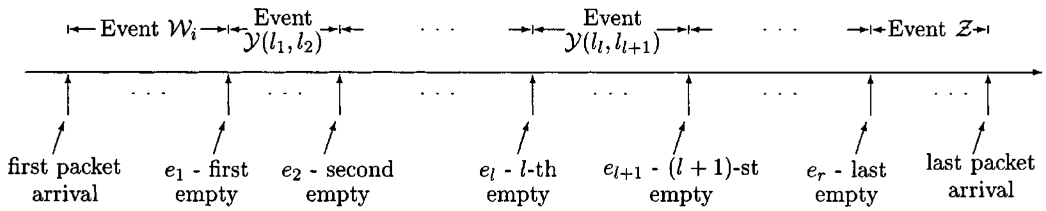

Fig. 3. A typical path for which the buffer becomes empty r times.

In order to continue, we need to take into account also those paths that do cause the buffer to become empty. Note that a path that starts when there are ι packets in the buffer, ends with μ packets in the buffer, and contains η events can become empty only if $\eta \geq \iota + \mu$ and at most $\mathcal{R} = 1 + (\eta - \iota - \mu)/2$ times ($\eta - \iota - \mu$ is obviously an even number). Furthermore, the first time the buffer can become empty is after ι events (all must be departures) and the last time it can become empty after $\eta - \mu$ events (and all the rest μ events must be arrivals). Also, the buffer can become empty only after events $\iota, \iota+2, \ldots, \iota+2(\mathcal{R}-1)$ (all must be departures). Let $\mathcal{E}$ be the set of these events, i.e., $\mathcal{E} = \{\iota, \iota+2, \ldots, \iota+2(\mathcal{R}-1)\}$. Consider now paths that the buffer becomes empty exactly r $(1 \leq r \leq \mathcal{R})$ times along the path. Denote the lth event that causes the buffer to become empty by $e_l (1 \leq l \leq r, e_l \in \mathcal{E}, e_1 < e_2 < \cdots < e_r)$. We note that such paths can be decomposed into three types of mutually exclusive events as follows.

Event $\mathcal{W}_\iota(e_1)$: the first time the buffer becomes empty is upon e_1 given that the path started with ι packets in the buffer.

Event $\mathcal{Y}(e_l, e_{l+1})$: the buffer becomes empty upon e_{l+1} after becoming empty upon e_l and was not empty in between the two events.

Event $\mathcal{Z}(e_r)$: upon e_r the buffer becomes empty for the last time.

Clearly, a path with the above properties is a succession of events $\mathcal{W}_\iota(e_1), \mathcal{Y}(e_1, e_2), \mathcal{Y}(e_2, e_3), \ldots, \mathcal{Y}(e_{r-1}, e_r), \mathcal{Z}(e_r)$ (see Fig. 3).

Let $w_\iota(e_1)$, $y(e_l, e_{l+1})$, and $z(e_r)$ be the probabilities of events $\mathcal{W}_\iota(e_1)$, $\mathcal{Y}(e_l, e_{l+1})$, and $\mathcal{Z}(e_r)$, respectively. It is clear that the probability that such a path will cause the buffer to become empty once is $\sum_{e_1 \in \mathcal{E}} w_\iota(e_1) z(e_1)$, twice

$$\sum_{e_1 \in \mathcal{E}} \sum_{e_2 \in \mathcal{E}, e_2 > e_1} w_\iota(e_1) y(e_1, e_2) z(e_2)$$

and, in general, r times

$$\sum_{e_1 \in \mathcal{E}} \sum_{e_2 \in \mathcal{E}} \cdots \sum_{e_r \in \mathcal{E}} w_\iota(e_1) y(e_1, e_2) \cdots y(e_{r-1}, e_r) z(e_r) \quad (16)$$

where we used the fact that $y(e_l, e_{l+1}) = 0$ when $e_l \geq e_{l+1}$ for $1 \leq l \leq r-1$. Equation (3) can be written in a matrix form

$$\boldsymbol{W}_\iota \cdot \boldsymbol{Y}^{r-1} \cdot \boldsymbol{Z}^T$$

where $\boldsymbol{W}_\iota$ and $\boldsymbol{Z}$ are $\mathcal{R}$-length row vectors with elements $w_\iota(\cdot)$ and $z(\cdot)$, respectively, and $\boldsymbol{Y}$ is an $\mathcal{R} \times \mathcal{R}$ matrix with elements $y(\cdot, \cdot)$. The elements of these vectors and matrix can be easily computed using (12). For instance, the lth $(0 \leq l \leq \mathcal{R}-1)$ element of the vector $\boldsymbol{W}_\iota$ is the probability that this part of the path starts with ι packets in the buffer, ends with an empty buffer after $\iota + 2l$ events, and the buffer does not become empty in between. By definition, this quantity is just $\xi_{\iota+2l}(\iota, 0)$. Therefore,

$$\boldsymbol{W}_\iota = \left(\xi_\iota(\iota, 0), \xi_{\iota+2}(\iota, 0), \ldots, \xi_{\iota+2(\mathcal{R}-1)}(\iota, 0)\right).$$

Similarly, the lth $(0 \leq l \leq \mathcal{R}-1)$ element of the vector $\boldsymbol{Z}$ is the probability that this part of the path starts with no packets in the buffer, ends with μ packets in the buffer after $\eta - \iota + 2l$ events, and the buffer does not become empty in between. By definition, this quantity is just $\xi_{\eta-\iota+2l}(0, \mu)$. Therefore,

$$\boldsymbol{Z} = (\xi_{\eta-\iota}(0, \mu), \xi_{\eta-\iota-2}(0, \mu), \ldots, \xi_{\eta-\iota-2(\mathcal{R}-1)}(0, \mu)).$$

Finally, the $l_1, l_2 (0 \leq l_1, l_2 \leq \mathcal{R}-1)$ element of the matrix $\boldsymbol{Y}$ is the probability that this part of the path starts and ends with no packets in the buffer after $2(l_2 - l_1)$ events and the buffer does not become empty in between. By definition, this quantity is just $\xi_{2(l_2-l_1)}(0, 0)$. Therefore,

$$\boldsymbol{Y} = \begin{pmatrix} 0 & \xi_2(0,0) & \xi_4(0,0) & \cdots & \xi_{2(\mathcal{R}-1)}(0,0) \\ 0 & 0 & \xi_2(0,0) & \cdots & \xi_{2(\mathcal{R}-2)}(0,0) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & \xi_2(0,0) \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Note that the matrix $\boldsymbol{Y}$ is very simple in the sense that each of its rows (except the first) is identical to the previous row with the elements moved one place to the right. Therefore, the computation of $\boldsymbol{Y}^j$ needs only $O(j \cdot n^2)$ operations.

To obtain $\zeta_\eta(\iota, \mu)$ we just need to sum over all paths that cause the buffer to become empty $r (0 \leq r \leq \mathcal{R})$ times, i.e.,

$$\zeta_\eta(\iota, \mu) = \xi_\eta(\iota, \mu) + \sum_{r=1}^{\mathcal{R}} \boldsymbol{W}_\iota \boldsymbol{Y}^{r-1} \boldsymbol{Z}^T.$$

The computation of $\zeta_\eta(\iota, \mu)$ needs $O(\mathcal{R}^2 \cdot n^2)$ operations. Once $\zeta_\eta(\iota, \mu)$ is known for all parameters, we use (8) and (10) to compute the vectors $\boldsymbol{V}_i$ and $\boldsymbol{U}$, respectively, and (9) to compute the matrix $\boldsymbol{S}$. Finally, (7) is used to compute $P_i(j, n)$ (recall that $P_i(0, n) = 1 - \sum_{j=1}^{n} P_i(j, n)$).

Having completed the computation of $P_i(j, n)$ we note that

$$P_i(\geq j, n) = \boldsymbol{V}_i \cdot \boldsymbol{S}^{j-1} \cdot \boldsymbol{e}^T$$

where $\boldsymbol{e}$ is a unit row vector, since event $\mathcal{U}$ is irrelevant when we are interested in at least j losses. The above computation needs $O(\mathcal{R}^2 \cdot n^3)$ operations. Finally, similarly to the reasoning for obtaining $P(\geq j, n)$ for small blocks (see Section III–A2), in the current situation we have that

$$P(\geq j, n) = \Pi(m)(\boldsymbol{e} \cdot \boldsymbol{S}^{j-1} \cdot \boldsymbol{e}^T - \boldsymbol{e} \cdot \boldsymbol{S}^j \cdot \boldsymbol{e}^T)$$

Note that $\boldsymbol{e} \cdot \boldsymbol{S}^{j-1} \cdot \boldsymbol{e}^T$ is just the sum of the elements of the matrix $\boldsymbol{S}^{j-1}$ and equals $\sum_{l_1, l_h} \mathrm{Prob}[X^j_{l_1, l_h}]$ where the set $X^j_{l_1, l_h}$ was defined in Section III-A2.

References

[1] I. Cidon, A. Khamisy, and M. Sidi, "Analysis of packet loss processes in high-speed networks," *IEEE Trans. Inform. Theory*, vol. 39, pp. 98–108, Jan. 1993.

[2] E. Altman and A. Jean-Marie, "Loss probabilities for messages with redundant packets feeding a finite buffer," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 778–787, June 1998.

[3] L. Takacs, *Combinatorial Methods in the Theory of Stochastic Processes*. New York: Wiley, 1967.

[4] P. Humblet, A. Bhargava, and M. G. Hluchyj, "Ballot theorems applied to the transient analysis of nD/D/1 queues," *IEEE/ACM Trans. Networking*, vol. 1, pp. 81–95, Feb. 1993.

[5] I. Cidon, R. Guerin, I. Kessler, and A. Khamisy, "Analysis of a statistical multiplexer with generalized periodic sources," *Queueing Syst.: Theory and Applic.*, vol. 20, pp. 139–169, 1995.