Comm. Korean Math. Soc. **13** (1998), No. 3, pp. 635–642

BLOCK LOSS DISTRIBUTION IN AN M/M/1 QUEUE WITH A CELL DISCARDING SCHEME

GYEMIN LEE, MIN-KON KWAG, JONGWOO JEON, AND CHONGKWON KIM

ABSTRACT. When an integrated communication system is congested, we may reserve some spaces for non-realtime traffic by discarding a part of realtime traffic. That is sensible because realtime traffic is insensitive to a few losses. Several discarding schemes have been developed including Separate Queue (SQ). Under such schemes, the block loss distribution, i.e., the distribution of the number of losses within a given block which consists of successive data of a type, is important. We derive the block loss distribution of the SQ scheme and modifies the SQ scheme with a threshold.

1. Introduction

Future networks must support a wide range of communication services such as video, voice, and data applications. Due to the transmission delay requirement, these services can be classified into realtime traffic and non-realtime traffic. If a realtime cell (or data) is not transmitted within its deadline, it is considered lost. On the other hand, non-realtime cells want to reach their destination without any losses. Therefore we may reserve some spaces for non-realtime cells by discarding realtime ones.

Various cell discarding schemes have been developed including PBS, Push-out, and Separate Queue (SQ) [2] [3]. The PBS scheme allows a realtime cell to access the buffer only if less than a given threshold T buffer places are occupied. Under the Push-Out scheme, a non-realtime cell replaces a buffered realtime cell when the buffer is full. The SQ scheme

Received August 30, 1997. Revised January 13, 1998.

1991 Mathematics Subject Classification: 60K25, 68M20.

Key words and phrases: realtime traffic, non-realtime traffic, single cell loss, block loss, two-level overload control.

This research was supported by SNU Daewoo Fund, 1995.

divides the buffer into two parts for each type traffic, where cells in the non-realtime buffer are served only if the realtime buffer is empty.

Since the quality of service of realtime traffic is sensitive to consecutive cell losses rather than a single cell loss, a proper performance measure for realtime traffic is the probability of consecutive cell losses. To derive this probability, many researchers have used the *independence assumption*, the events of cell losses are mutually independent and identical. However, Cidon *et al.* [1] showed that there is a high correlation between consecutive cell losses. Thus we consider the block loss distribution, i.e., the distribution of the number of losses within a given block which consists of successive cells of a type. For the system either with the PBS scheme or the Push-out scheme, its block loss distribution has already been derived in [4] and [5].

In this paper, we derive the block loss distribution in the SQ scheme. And we suggest a new cell discarding scheme, namely the Separate Queue with Threshold (SQT) scheme, which prevents too excessive cell losses of non-realtime traffic in the SQ scheme by introducing a threshold. When the number of cells in the non-realtime buffer exceeds T, the SQT scheme transmits non-realtime cells until there are less that T cells in the buffer. The loss distribution for the SQT scheme is also derived.

The remainder of this paper is constructed as follows. Section 2 describes the system model and defines several notations. In Section 3, we explains the recursive algorithm to compute the block loss distributions for the SQ scheme and the SQT scheme. Some numerical studies are presented in Section 4. Finally Section 5 contains some conclusion.

2. Modeling and notations

Realtime and non-realtime cells are assumed to arrive according to Poisson processes with rates λ^r and λ^n, respectively. For simple analysis, we assume that transmission (service) times are exponentially distributed with a common rate μ for either type cells. Changing the service time distribution from deterministic to exponential has no significant effect on the queueing process. See [6] and [8]. Let N_1 and N_2 denote sizes of realtime and non-realtime buffers, respectively. A threshold value in the SQT scheme is denoted by T.

To investigate the behavior of cell losses, we group consecutive cells generated by the same source into blocks. The block size can be arbitrarily determined. Before computing the block loss distribution, we introduce several notations.

- $\Pi^r(i)$(or $\Pi^n(i)$), $i = 0, 1, \ldots, N_1$(or N_2) : Probabilities that i realtime(or non-realtime) cells exist in the system when a new cell arrives.
- $Q_i^r(k)$(or $Q_i^n(k)$), $i = 0, 1, \ldots, N_1$(or N_2), $0 \leq k \leq i$: Probabilities that k cells out of i realtime(or non-realtime) cells in the system are transmitted during an inter-arrival period.
- $P^r(j, n)$(or $P^n(j, n)$), $n \geq 1$, $0 \leq j \leq n$: Probabilities that j cells in a realtime(or non-realtime) block of size n are lost.
- $P_i^r(j, n)$(or $P_i^{\bar{r}}(j, n)$), $i = 0, 1, \ldots, N_1$, $n \geq 1$, $0 \leq j \leq n$: Probabilities that j losses occur in the realtime block of size n, given that i cells exist in the system just before the arrival of a realtime(or non-realtime) cell.
- $P_i^n(j, n)$(or $P_i^{\bar{n}}(j, n)$), $i = 0, 1, \ldots, N_2$, $n \geq 1$, $0 \leq j \leq n$: Probabilities that j losses occur in the non-realtime block of size n, given that i cells exist in the system just before the arrival of non-realtime(or realtime) cell.
- $p(r)$(or $\bar{p}(r)$) : Probability that an arriving cell is a realtime (or non-realtime) cell.

3. Block loss distribution

3.1. Separate queue scheme

Since the realtime buffer has priority in service, the model for the realtime buffer is an $M/M/1/N_1$ queue. Therefore $\Pi^r(i), 0 \leq i < N_1$, are given by

$$\Pi^r(i) = \frac{\rho_r^i}{\sum_{j=0}^{N_1} \rho_r^i} \quad \text{where} \quad \rho_r = \frac{\lambda^r}{\mu}. \tag{1}$$

Our purpose is to compute $P^r(j,n)$ and $P^n(j,n)$. Using the conditional probabilities $P_i^r(j,n)$ and $P_i^n(j,n)$, we get

$$P^r(j,n) = \sum_{i=0}^{N_1} \Pi^r(i) P_i^r(j,n), \tag{2}$$

$$P^n(j,n) = \sum_{i=0}^{N_2} \Pi^n(i) P_i^n(j,n). \tag{3}$$

First consider a way to obtain $P_i^r(j,n)$ recursively. The initial values of $P_i^r(j,n)$ are as follows.

$$P_i^r(j,1) = \begin{cases} 1 & , j = 0 \\ 0 & , j \geq 1 \end{cases}, \qquad 0 \leq i < N_1, \tag{4}$$

and

$$P_i^r(j,1) = \begin{cases} 1 & , j = 1 \\ 0 & , j = 0, j \geq 2 \end{cases} \qquad i = N_1. \tag{5}$$

Now consider general cases $P_i^r(j,n)$ for $n \geq 2$. If the first cell in a realtime block arrives and sees i cells($0 \leq i < N_1$) in the system, it can enter the system. As a result there are $i+1$ cells in the system. Thus j cells out of the next $(n-1)$ realtime cells must be lost. Using $Q_i^r(k)$, $p(r)$, $p(\bar{r})$ defined in the previous section, we get

$$P_i^r(j,n) = \sum_{k=0}^{i+1} Q_{i+1}^r(k)[p(r)P_{i+1-k}^r(j,n-1) + p(\bar{r})P_{i+1-k}^{\bar{r}}(j,n-1)] \tag{6}$$

for $0 \leq i < N_1$. On the other hand, the first cell in the realtime block will be lost if there are more than N_1 cells in the system. Consequently $(j-1)$ cells out of the next $(n-1)$ realtime cells must be lost. Thus we have

$$P_i^r(j,n) = \sum_{k=0}^{i} Q_i^r(k)[p(r)P_{i-k}^r(j-1,n-1) + p(\bar{r})P_{i-k}^{\bar{r}}(j-1,n-1)] \tag{7}$$

for $i = N_1$. From the fact that any non-realtime cells cannot enter the realtime buffer under this scheme, we get

$$P_i^{\bar{r}}(j,n) = \sum_{k=0}^{i} Q_i^r(k)[p(r)P_{i-k}^r(j,n) + p(\bar{r})P_{i-k}^{\bar{r}}(j,n)] \tag{8}$$

for $0 \leq i \leq N_1$. It is clear that $p(r) = \lambda^r/(\lambda^r + \lambda^n)$ and

$$Q_i^r(k) = \begin{cases} s^k a & , 0 \leq k < i \\ s^i & , k = i \end{cases} \tag{9}$$

where $s = \mu/(\mu + \lambda^r + \lambda^n)$ and $a = (\lambda^r + \lambda^n)/(\mu + \lambda^r + \lambda^n)$. Through the above equations (4)–(9) and (2), we can calculate $P_i^r(j,n)$ recursively.

Similarly we can calculate $P_i^n(j,n)$ if we know $\Pi^n(i)$ and $Q_i^n(k)$. The service rate for non-realtime traffic is regulated according to the state of realtime buffer. If the number of cells in the realtime buffer is defined as a phase variable, the service distribution for non-realtime traffic is a *phase(PH) distribution* [7]. Therefore the model for the non-realtime buffer is an $M/PH/1/N_2$ queue. From the results in [8], we can obtain $\Pi^n(i)$.

Since cells in the realtime buffer are transmitted precedently, $Q_i^n(k)$ depends on the number of cells in the realtime buffer. Let $Q_{il}^n(k)$ be the probabilities that k cells out of i cells in the non-realtime buffer are transmitted during an inter-arrival period, given that l cells exist in the realtime buffer. Then we have

$$Q_i^n(k) = \sum_{l=0}^{N_1} \Pi^r(l) Q_{il}^n(k) \tag{10}$$

It is clear that $Q_{il}^n(k)$ are

$$Q_{il}^n(k) = \begin{cases} \sum_{j=0}^{l} s^j a & , k = 0, \\ s^{k+l} a & , 1 \leq k < i, \\ s^{i+l} & , k = i. \end{cases} \tag{11}$$

Thus we can calculate $P_i^n(j,n)$ through (10), (11) and (3) recursively.

3.2. SQ with threshold

We shall use notations defined in the previous subsection unless explicitly stated. For realtime traffic, the equations for $P_i^r(j,n)$, $P_i^{\bar{r}}(j,n)$, $P_i^n(j,n)$, and $P_i^{\bar{n}}(j,n)$ are the same as those in the SQ mechanism if $Q_i^r(k)$, $Q_i^r(k)$, $\Pi^r(i)$, and $\Pi^n(i)$ are given.

When occupancy of the non-realtime buffer exceeds threshold T, non-realtime cells change into high-priority cells. As a result, the service rate for realtime traffic are modulated with the number of cell in the non-realtime buffer. From similar arguments in the previous subsection, we

know that the model for the realtime buffer is an $M/PH/1/N_1$ queue. The procedure suggested in [8] enables us to calculate $\Pi^r(i)$ and $\Pi^n(i)$ as follows.

$$\Pi^r(i) = \sum_{j=0}^{N2} \Pi_{ji}, \ 0 \le i < N_2, \tag{12}$$

$$\Pi^n(j) = \sum_{i=0}^{N1} \Pi_{ji}, \ 0 \le j < N_1, \tag{13}$$

where Π_{ij} is the joint distribution of the queue length and the phase in the $M/PH/1/N_1$ queue.

Let $Q^r_{il}(k)$ be the probability that $k(0 \le k \le i)$ cells out of i cells in the realtime buffer are transmitted during an inter-arrival period, given that l cells are in the non-realtime buffer. Then we have

$$Q^r_i(k) = \sum_{l=0}^{N_2} \frac{\Pi_{il}}{\Pi^n(l)} Q^r_{il}(k). \tag{14}$$

When there are less than T cells in the non-realtime buffer, realtime cells must be transmitted precedently. Thus $Q^r_{il}(k), \ l \le T$ are given by

$$Q^r_{il}(k) = \begin{cases} s^k a & , 0 \le k < i, \\ s^i & , k = i \end{cases} \tag{15}$$

If the number of cells in the non-realtime buffer exceeds T, realtime traffic can not be served until non-realtime buffer occupancy is less than T. Therefore $Q^r_{il}(k), \ l > T$ are given by

$$Q^r_{il}(k) = \begin{cases} \sum_{j=0}^{l-T} s^j a & , k = 0 \\ s^{l-T+k} a & , 0 < k < i \\ s^{l-T+i} & , k = i. \end{cases} \tag{16}$$

Now consider $Q^n_{il}(k)$ defined in the previous subsection. When i is less than T, non-realtime cells can not be served until the realtime buffer is empty. For $i > T$, $(i - T)$ non-realtime cells above the threshold, realtime cells, and non-realtime cells below the threshold are transmitted sequentially due to preemptive service. This fact states that $Q^n_{il}(k), i > T$

are given by

$$Q_{il}^{n}(k) = \begin{cases} a & ,k = 0, \\ s^k a & ,k < i - T, \\ \sum_{j=0}^{l} s^{i-T+j} a & ,k = i - T, \\ s^{k+l} a & ,k > i - T, \\ s^{i+l} & ,k = i. \end{cases} \tag{17}$$

4. Numerical results

For numerical studies, we assume that the block size is 10 and the buffer is equal to 20. And the block having more than 3 cell losses is assumed to be lost. We divide the buffer of equal sizes for the SQ mechanism and the SQT mechanism. The threshold value in the SQT mechanism is 8. Let us suppose that the two traffic types have the following quality of service requirements.

(1) For realtime traffic, the probability of a block loss should be less than 10^{-3}.
(2) For non-realtime traffic, the average of block loss distribution should be less than 10^{-6}.

Consider an admissible load based on the above requirements. We use the recursive algorithms suggested in [4] and [5] for the block loss distributions of the PBS scheme and the Push-out scheme. Table 1 shows that the Push-Out mechanism is superior to other mechanisms in terms of admissible loads regardless of the traffic ratio. Because the Push-Out mechanism fully utilizes the buffer space. The performance of the PBS and Push-Out mechanisms seems constant with respect to the traffic ratio. The SQT mechanism is better than the SQ scheme in terms of admissible loads. It is because the SQT scheme protects overflow in the non-realtime buffer by introducing a threshold.

Table 1. Admissible Load with respective to traffic ratios

Schemes	$\lambda^r/\lambda^n = 1/3$	$\lambda^r/\lambda^n = 1/1$	$\lambda^r/\lambda^n = 3/1$
PBS	0.66	0.67	0.67
Push-out	0.69	0.72	0.72
SQ	0.10	0.10	0.40
SQT	0.30	0.57	0.72

5. Conclusion

In this paper, we derived a way to obtain the block loss distribution within a fixed size block under the SQ scheme and suggested a new cell discarding scheme, namely a SQT mechanism. Also we presented several numerical results about admissible load. These result showed that the Push-Out scheme is better than any other scheme in terms of admissible loads and that the SQT mechanism gives an improvement on the performance of the SQ scheme.

References

[1] I. Cidon, A. Khamish, and M. Sidi, *Analysis of packet loss processes in high-speed networks*, IEEE Trans. on Information Theory **39** (1993), 98-108.

[2] H. Kröner, *Comparative study of space priority mechanism for ATM networks*, IEEE INFOCOM '90 (1990), 1136-1143.

[3] H. Kröner, G. Hebuterne, and P. Boyer, *Priority management in ATM switching node*, IEEE J. Selected Areas in Communications **9** (1991), 418-427.

[4] M. K. Kwag. and C. K. Kim, *Space priority Queueing Stragy*, ICCN Conference (1993), 77-84.

[5] M. K. Kwag. and C. K. Kim, *An analysis of loss process in an ATM network under Partial Buffer Sharing policy*, J. of Korean Institute of Comm. Sci. **19** (1994), 2328-2339.

[6] S. Q. Li, *Overload Control in a finite Message Sorage Buffer*, IEEE trans. Comm. **37** (1989), 1330-1338.

[7] M. F Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Application*, Marcel Dekker Inc., 1989.

[8] J. Ye and Li S. Q., *Folding Alogorithm: A Computational Method for Finite QBD Process with Level-Dependent Transitions*, IEEE trans. on comm. **42** (1994), 625-639.

Gyemin Lee, Min-Kon Kwag and Jongwoo Jeon
Department of Statistics
Seoul National University
Seoul 151-742, Korea

Chongkwon Kim
Department of Computer Science
Seoul National University
Seoul 151-742, Korea