Performance analysis of a dynamic priority scheduling method in ATM networks

T.-Y. Huang J.-L.C. Wu

1

Indexing terms: ATM networks, Congestion control, Dynamic priority, Broadband ISDN

Abstract: Broadband ISDN using asynchronous transfer mode as the transport mode carries traffic of several different classes, including one or more real-time traffic classes, each with its own set of traffic characteristics and performance requirements. To support multiple classes of traffic in ATM networks, multiple priority levels are provided, and different priority levels are given to different classes of traffic. A dynamic priority scheme is proposed as an efficient scheduling method at a switching node in an ATM network serving multiple classes of delay-sensitive traffic. The advantage of this protocol lies in that, by properly adjusting the control parameters, it can meet the average delay requirement for different classes and thus guarantee the quality of service. In addition, each class of service can be guaranteed a minimum of bandwidth, which can prevent the low priority class from starvation.

1 Introduction

Broadband ISDNs (B-ISDN) using the asynchronous transfer mode (ATM) [1] are expected to support a range of voice, video, data, image, and multimedia services using available resources. To achieve the goals of supporting diverse services and traffic mixes and efficient use of network resources, the design of a congestion control becomes an important challenge [2-12].

Network congestion controls can be implemented at a number of levels in an ATM network, including call-level control, networkwide control at the ATM level and switch-element internal control [13]. For each call of the call-level control, there should be a service contract, specifying a set of service parameters between the network and the terminals. The service parameters are set during the call establishment procedure, which put limits on the traffic between the network and terminals.

With service parameters established at the call level, there must be a set of mechanisms for observing these service agreements during the transport of ATM cells. This is one of the functions of the networkwide ATMlevel control. Another function of the networkwide ATM-

Tien-Yu Huang is with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

IEE PROCEEDINGS-I, Vol. 140, No. 4, AUGUST 1993

level control is to provide congestion status information to end terminals. Actions in networkwide ATM-level control include traffic monitoring, selective cell discard, and triggering the backward notification of congestion.

Internal control, also known as cell-level control, is the ability for a switch element to perform service scheduling for bandwidth and buffers consistent with the service classes supported by the switch. This could involve buffer allocation between service classes and real-time allocation of bandwidth for each service class and thus prevent one service class from starving for transmission. This establishes a system of relative priorities between classes.

Various priority schemes can be used as a scheduling method at a switching node in an ATM network. The simplest priority scheme is the static priority scheme. In this, priority is always given to the more delay-sensitive class. This scheme frequently causes starvation for the less delay-sensitive traffic.

To overcome the drawbacks of static priority scheme, a dynamic priority scheme is needed. We propose and analyse a new dynamic priority scheme that is suitable for an ATM switching network. In our proposal, each class of service can be guaranteed a minimum of bandwidth, which can prevent the low-priority service class from starvation.

2 Protocol description

The main reason for imposing a priority discipline on arrivals is to provide preferential treatment to 'high priority' group, e.g. voice cells in ATM networks, at the expense of 'low priority' groups, e.g. data cells. Research on priority schemes has been concerned mainly with static priority assignment. However, they lack flexibility and cannot handle the problem of starvation for low priority groups. So there is a need to propose dynamic priority schemes to be able to adjust the relative priorities of packets according to their required delay limits in the system.

2.1 Dynamic priority protocols

Kleinrock [14-18] proposed a time-dependent priority queueing model which consists of P priority groups associated with a set of variable parameter b_p where $0 \le b_1 \le b_2 \le \cdots \le b_p$. The instantaneous priority at time t of a group i customer that arrived at time τ is simply $q_i(t) = (t - \tau)b_i$.

Another dynamic priority discipline, called head-ofthe-line with priority jumps (HOL-PJ), was proposed by Lim and Kobza [19] in 1988. They assume C classes of packets with separate queues for each class. Packets are FCFS within each queue, while queue *i* has nonpreemptive priority over queue *j* if i < j. However, packets have maximum limit on the waiting time in queue. When

[©] IEE, 1993

Paper 94551 (E7), first received 1 July 1992 and in revised form 22 December 1992

Jean-Lien Wu is with the Computer Network Laboratory, Department of Electronic Engineering, National Taiwan Institute of Technology, 43 Keelung Road, Section 4, Taipei, Taiwan, Republic of China

the limit expires, that packet moves to the end of the next higher priority queue. Thus the queueing time at each node will be improved.

Chen et al. [20] proposed two schemes of dynamic priority: the 'oldest customer first' (OCF) discipline and the 'earliest deadline first' (EDF) discipline. They assumed multiple queues in the system. Since arrivals that happen to join the longer queue would experience a longer delay time, it is not fair to pick customers for service from queue to queue. Rather, it would be more appropriate to give preference to those customers that have spent more time in the network. This is the so-called OCF discipline. The EDF discipline gives service to the customer with the earliest deadline. In fact, it is a generalisation of the OCF discipline [21].

2.2 Priority promotion protocol

A possible disadvantage of these priority schemes is the processing overhead required for monitoring cells for priority change. Also, each arriving cell needs to be time stamped. In an ATM network, the processing time should be as small as possible, because the transmission capacity is very high. In this section, we describe a priority promotion protocol, which can be used as a scheduling method at a switching node in an ATM network because of its easy implementation and good performance.

Consider an ATM switching node consisting of N classes of cells numbered from 1 to N. Class i cells have higher priority than those of class j if i < j. An arriving cell joins the tail of its corresponding class of queue. The highest priority cell is chosen to be transmitted. The access time of a cell is defined as the interval from the time the cell becomes the head of its joining queue to the time that it is chosen for transmission. A limit L is imposed on the access time of each class of cells. If the access time of a class *i* cell exceeds limit L_i , it promotes its priority so that it is higher than cells of other classes and can be chosen for transmission in the next time slot. This protocol can be implemented by employing a counting register at each queue. If the head cell in queue i waits one time slot, the counting register of class i increases one. When the counting register reaches the limit L_i , the priority of the head-of-line cell of class *i* will be promoted and the priority-promoted cell is entitled to be transmitted ahead of other head-of-line cells which do not promote their priorities. The counting register of class i is reset to zero after the head-of-line cell of class i is transmitted. If more than two classes of cells promote their priority simultaneously, the lower priority class (class j) will be chosen for transmission ahead of the higher priority class (class i).

In our proposed priority promotion scheme, a minimum amount of bandwidth is guaranteed by choosing an appropriate value for the limit L_i . Consider that there are N classes of traffic in an ATM network. When the traffic of all classes are heavy, there are always higher priority cells ahead of class N cells. A class N cell can be transmitted only when it promotes its priority. The priority of the cells of class N is promoted every $L_N + 1$ time slots, thus the allocated bandwidth of class N is $1/(L_N + 1)$ of the total bandwidth. The remaining part of the total bandwidth is equal to $L_N/(L_N + 1)$, to be shared by other classes of cells. For a class N - 1 cell, there are always higher priority cells ahead of it, thus the cells of class N - 1 will promote the priority every $L_{N-1} + 1$ time slots. The allocated bandwidth of class N - 1 is $1/(L_{N-1} + 1)$ of the total bandwidth if $1/(L_{N-1} + 1) \leq$

 $L_N/(L_N + 1)$. In other words, if $1/(L_{N-1} + 1) + 1/(L_N + 1) \le 1$, the minimum amount of bandwidth of class N-1 is $1/(L_{N-1} + 1)$ of the total bandwidth. Similarly, one can easily induce that class *i* traffic can be allocated $1/(L_i + 1)$ of the total bandwidth if

$$\sum_{\leq j \leq N} 1/(L_j + 1) \leq$$

From this description, one concludes that if the network traffic is heavy for each traffic class, class *i* traffic can be at least allocated $1/(L_i + 1)$ of the bandwidth, if

$$\sum_{\leq j \leq N} 1/(L_j + 1) \leq 1$$

If the parameter L_i is appropriately chosen the minimum bandwidth requirement for each traffic class is achievable.

3 Waiting time analysis

In this Section, we analyse the average waiting time for different traffic classes under priority promotion discipline and make some additional simplifying assumptions that are different from those made in Section 2. Assume that there is a single cell buffer for each traffic class. The buffer is released at the beginning of a cell transmission. A class *i* cell arrives at buffer *i* with probability λ_i during a slot duration. Cell arrivals will be considered at slot boundaries only, namely, just before a slot end. Depending on the availability of the buffer *i*, a class *i* cell arriving at the buffer will be either accepted or blocked.

The analytical model is designed as a Markov chain in which state transitions occur at the slot boundaries. The state space can be described by the vector $(n_1, n_2, ..., n_N)$, where

if has form the same star

n _i	=		

,	I build t is emply
l I	if a class <i>i</i> cell just arrives at buffer <i>i</i>
2	if a cell in buffer <i>i</i> has waited one time slot
3	if a cell in buffer <i>i</i> has waited two time slots
L _a + 1	if a cell in buffer <i>i</i> has waited no less than

i + 1 if a cell in buffer *i* has waited no less than L_i time slots, and has promoted its priority

The number of possible states for a class *i* cell are $L_i + 2$, thus the total number of possible states in the Markov chain is $(L_1 + 2), (L_2 + 2), \ldots, (L_N + 2)$. It is tedious work to find all possible states and the transition probabilities between states. In Section 7.1 we present the rules that can be formulated to generate the state space and the transition probabilities. A detailed procedure used to generate the state space and the transition probability matrix is presented in Section 7.2.

Let Q be the state transition probability matrix of the system generated from the procedure, and π be the steady-state probability vector. The steady-state probability vector π can be obtained by solving the state balance equation:

$$\pi \boldsymbol{Q} = \pi \tag{1}$$

and the normalisation equation

$$\sum_{\substack{n_1, n_2, \dots, n_N}} \pi(n_1, n_2, \dots, N) = 1$$
 (2)

The expected number of class *i* cells is equal to the sum of state probability $\pi(n_1, n_2, ..., n_N)$, where $n_i \neq 0$, or

$$E(n_i) = \sum_{n_i \neq 0, \text{ all } n_1, n_2, \dots, n_N} \pi(n_1, n_2, \dots, n_N)$$
(3)

IEE PROCEEDINGS-I, Vol. 140, No. 4, AUGUST 1993

286

The throughput of class *i* traffic is equal to the sum of state probability $\pi(n_1, n_2, ..., n_N)$ where a class *i* cell is chosen to be transmitted, or

$$S_{i} = \sum_{\substack{n_{i} \neq 0, n_{i} \neq L_{i} + 1 \\ n_{i} = 0, g < i \\ n_{k} \neq L_{k} + 1, k > i}} \pi(n_{1}, n_{2}, ..., n_{N}) + \sum_{\substack{n_{i} = L_{i} + 1 \\ n_{i} \neq L_{k} + 1, k > i}} \pi(n_{1}, n_{2}, ..., n_{N})$$
(4)

The delay time W^i of a class *i* cell is defined as the interval from the time a cell arrives at queue *i* to the time that the cell is transmitted. W^i can be calculated by Little's formula

$$S_i W^i = E(n_i) \tag{5}$$

The waiting time W_q^i is equal to the delay time W^i minus a cell transmission time, thus

$$W_a^i = W^i - 1 \tag{6}$$

4 Analytical and simulation results

Figs. 1a and 1b depict the waiting time against arrival rate of class 1 where the access time limit L for each class



class 2
is 3. The arrival rates of classes 2 and 3 are fixed at some values, while the arrival rate of class 1 varies from 0.1 to 1.0. When the load of class 1 gets heavy, the waiting time of class 3 increases slowly, while the waiting time of class 2 increases quickly. Therefore, the waiting time of low

priority classes are less affected by the increased arrival

IEE PROCEEDINGS-I, Vol. 140, No. 4, AUGUST 1993

rate of the highest priority class in priority promotion scheme.

Figs. 2a and 2b depict the waiting time versus arrival rate of class 3 when class 2 is heavily loaded. The arrival rates of classes 1 and 2 are 0.2 and 1.0 respectively, while



Fig. 2 Waiting time against arrival rate of class 3 when class 2 is heavily loaded

 $\begin{array}{l} a \ L_1 = L_2 = L_3 = 3 \\ b \ L_1 = 2, \ L_2 = 3, \ L_3 = 4 \\ class 1 \ arrival rate = 0.2 \\ class 2 \ arrival rate = 1.0 \\ \hline & \ simulation result \\ \hline & \ class 1 \\ \hline & \ class 1 \\ \hline & \ class 2 \\ \hline & \ class 3 \end{array}$

the arrival rate of class 3 varies from 0.1 to 1.0. In Fig. 2a, the access time limit for each class is 3. The waiting time of class 3 stays the same when the arrival rate λ_3 varies. Because the traffic of class 2 is heavy, a class 3 cell will not be chosen for transmission unless it promotes its priority. It must wait for an access time limit L_3 to promote its priority then it can be transmitted immediately, thus the waiting time of class 3 is bounded within three time slots. In Fig. 2b, the arrival rate of each class is the same as that in Fig. 2a, but the access time limit of each class is changed. The access time limits of classes 1, 2, and 3 are 2, 3, and 4, respectively. One observes the same behaviour as in Fig. 2a that the waiting time of class 3 is kept as L_3 .

Figs. 3a and 3b depict the waiting time versus arrival rate of class 3 when class 1 is heavily loaded. The arrival rates of classes 1 and 2 are 1.0 and 0.2, respectively, while the arrival rate of class 3 varies from 0.1 to 1.0. In Fig. 3a, the access time limit for each class is 3. The waiting time of class 3 stays the same and the waiting time of class 2 is slightly above 3. The reason is that when class 1 is heavily loaded, the cells of low priority classes can be transmitted only when they are priority promoted. The access time limits of classes 2 and 3 are all 3, whereas the priority of class 3 is higher than the priority of class 2 after they are both priority promoted. Thus, the waiting

287

time of class 3 traffic is bounded within the access time limit L_3 , while the waiting time of class 2 traffic is slightly longer than the access time limit L_2 . In Fig. 3b, the



Fig. 3 Waiting time against arrival rate of class 3 when class 1 is heavily loaded

 $a L_1 = L_2 = L_3 = 3$ $b L_1 = 2, L_2 = 3, L_3 = 4$ class 1 arrival rate = 1.0 class 2 arrival rate = 0.2 o simulation class _ _ _ _ class 2 class 2

288

arrival rate of each class is the same as that in Fig. 3a, but the access time limit of each class is changed. The access time limits of classes 1, 2, and 3 are 2, 3, and 4, respectively. One observes the same behaviour as in Fig. 3a that the waiting time of class 3 traffic is kept as L_3 , and the waiting time of class 2 traffic is slightly longer than L_2 .

In Figs. 4a and 4b, the access time limit of high priority class is adjusted to be smaller than that of low priority class. The access time limit of classes 1, 2, and 3 are 2, 3, and 4, respectively. The arrival rates for classes 2 and 3 are fixed at some values, and the arrival rate of class 1 varies from 0.1 to 1.0. It is found that the waiting time of class 2 is always smaller than that of class 3 in Figs. 4a and 4b. Thus, to keep the high priority class always performs better than the low priority class, the access time limit L of low priority class should be larger than that of high priority class.

In Table 1, the arrival rates for classes 1, 2, and 3 are all 1, and the access time limits for all classes are all 3. Table 1 demonstrates the bandwidth allocation when the load is heavy. One finds that classes 2 and 3 cells occupy $\frac{1}{4}$ of the total bandwidth, respectively, while class 1 cells occupy $\frac{1}{2}$ of the total bandwidth. Because the arrival rate of class 1 is 1, a class 3 cell can be transmitted only when it promotes its priority. The priority of class 3 is promoted every $L_3 + 1$ time slots, thus the bandwidth occupied by class 3 traffic is $1/(L_3 + 1)$ of the total bandwidth, which is $\frac{1}{4} = 0.25$. The remaining part of the total bandwidth is 0.75 to be shared by class 1 and class 2. For the same reason as that of class 3, the bandwidth occupied by class 2 traffic is $1/(L_2 + 1) = 0.25$. The remaining bandwidth is (1 - 0.25 - 0.25) = 0.5 which is allocated to class 1. Thus, when the arrival rate of high



Fig. 4 Waiting time against arrival rate of class 1

a class 2 arrival rate = 0.8

class 2 arrival rate = 0.2b class 2 arrival rate = 0.4

class 3 arrival rate = 0.4

simulation $L_1 = 2, L_2 = 3, L_3 = 4$

Table 1 : Bandwidth allocation for each class of traffic

Class number	Per cent of allocated bandwidth			
1	0.5			
2	0.25			
3	0.25			
$(L_1 = L_2)$	$= L_3 = 3, \lambda_1 =$	$A_2 = A_3 = 1$		

priority class is heavy, the minimum bandwidth of low priority class is preserved, so that the low priority class in the proposed priority scheme does not suffer from starvation.

5 Conclusion

A new dynamic priority scheme, the priority promotion discipline has been proposed for ATM networks that provide guaranteed quality of service. The access time of each class of service is imposed with a limit to prevent itself from waiting too long. To keep a higher priority class always performs better than a lower priority one, the access time limit L of lower priority class should be larger than that of higher priority class.

When the arrival rate of each traffic class is heavy, by appropriately choosing the access time limit, the starvation of low priority class can be avoided because the

IEE PROCEEDINGS-I, Vol. 140, No. 4, AUGUST 1993

access time limit guarantees a minimum amount of bandwidth to be allocated.

The waiting time of each class of service in relation to the access time limit is analysed using a Markov chain model. By properly adjusting the access time limit of each class in priority promotion protocol, the expected bandwidth of each class is achievable and the delay requirement is met for different classes, thus bring fairness of service and result in resource saving.

The implementation of the priority promotion protocol is quite simple and it is suitable for being a scheduling method in ATM networks.

6 References

1

- CCITT Proposed Recommendation I.311, June 1991
 BAE, J.J., and SUDA, T.: 'Survey of traffic control schemes and protocols in ATM networks', Proc. IEEE, 1991, 79, pp. 170-189
- 3 GERSHT, A., and LEE, K.J.: 'A congestion control framework for ATM networks'. Proceedings of IEEE Infocom '89, Ottawa, Canada, 1989, pp. 701-710
- 4 WOODRUFF, G.M., ROGERS, R.G., and RICHARDS, P.S.: 'A congestion control framework for high-speed integrated packetized transport'. Proceedings of IEEE Globecom '88, Hollywood, FL,
- tuansport: Proceedings of TEEE Cholecolin 38, Problemout, PL, USA, 1988, pp. 7.1.1–7.1.5
 5 HIRANO, M., and WATANABE, N.: 'Characteristics of a cell multiplexer for bursty ATM traffic'. Proceedings of IEEE ICC '89, Boston, MA, USA, 1989, pp. 13.2.1–13.2.5
 6 DITTMANN, L., and JACOBSEN, S.B.: 'Statistical multiplexing of DITTMANN', L., and JACOBSEN', S.B.: 'Statistical multiplexing of DITTMAN', L., and JACOBSEN', S.B.: 'S
- b) IT IMANN, L., and JACOBSEN, S.B.: Statistical multiplexing of identical bursty sources in an ATM network'. Proceedings of IEEE Globecom '88, Hollywood, FL, USA, 1988, pp. 39.6.1-39.6.5
 7 GALLASSI, G., RIGOLIO, G., and FRATTA, L.: 'ATM: band-width assignment and bandwidth enforcement policies'. Proceedings (URD 1) and the proceedings (URD 1) and the proceedings).
- of IEEE Globecom '89, Dallas, TX, USA, 1989, pp. 49.6.1-49.6.6 8 HUI, J.Y., and ARTHURS, E.: 'A broadband packet switch for integrated transport', *IEEE J. Sel. Areas Commun.*, 1987, **5**, pp. 1264-1273
- SRIRAM, K., and WHITT, W.: 'Characterizing superposition arrival processes in packet multiplexers for voice and data', *IEEE J*. Sel. Areas Commun., 1986, 4, pp. 833-846 10 NAKAMAKI, K., KAWAJATSU, M., and NOTOYA, A.: Traffic
- control for ATM networks'. Proceedings IEEE ICC '89, Boston,
- MA, USA, 1989, pp. 22.5.1–22.5.5
 SIDI, M., LIU, W.Z., CIDON, I., and GOPAL, I.: 'Congestion control through input rate regulation'. Proceedings IEEE Globecom '89, Dallas, TX, USA, 1989, pp. 49.2.1–49.2.5
 TURNER, J.S.: 'New directions in communications', IEEE Commun. Mag., October 1986, 25, pp. 8–15
 ECK PERG, A.F. DOSHI, P. and ZOCCOLILLO, P.: 'Control
- ECKBERG, A.E., DOSHI, B.T., and ZOCCOLILLO, R.: 'Control-ling congestion in B-ISDN/ATM: issues and strategies', IEEE
- Commun. Mag., Sept. 1991, 29, pp. 64–70
 14 KLEINROCK, L.: 'Queueing system, vol. II: computer application' (Wiley, New York, 1976)
- 15 KLEINROCK, L.: 'Queueing system, vol. I: theory' (Wiley, New York, 1975)
- 16 KLEINROCK, L.: 'Communication nets: stochastic message flow and delay' (McGraw-Hill, New York, 1964)
- 17 KLEINROCK, L.: 'A delay dependent queue discipline', Nav. Res. Logist. Q., 1964, 11, pp. 329-341
- KLEINROCK, L., and FINKLESTIEN, R.P.: Time dependent priority queues', Oper. Res., 1967, 15, pp. 104-116
 LIM, Y., and KOBZA, J.: 'Analysis of a delay-dependent priority
- discipline in a multiclass traffic packet switching node'. Proceedings of IEEE Infocom '88, New Orleans, LA, USA, 1988, pp. 9A.4.1-9A.4.1.10
- 20 CHEN, T.M., WALRAND, J., and MESSERSCHMITT, D.G.: Dynamic priority protocols for packet voice', *IEEE J. Sel. Areas Commun.*, 1989, 7, (5), pp. 632-643
 21 GOLDBERG, H.: 'Analysis of the earliest due data scheduling rule in queueing systems', *Math. Oper. Res.*, May 1977, 2, pp. 145-154

7 Appendix

7.1 State space and transition probabilities

To generate all possible states, start from an initial state (0, 0, ..., 0) and consider all possible events that can occur in that state. The next possible events of a state are independent combinations of all possible changes in each traffic class. Let the state at the kth time slot be (n_1^k, n_2^k) ,

IEE PROCEEDINGS-I, Vol. 140, No. 4, AUGUST 1993

 \dots, n_N^k , and the state at the (k + 1)th time slot be $(n_1^{k+1}, n_2^{k+1}, \dots, n_N^{k+1})$. The possible changes that can happen for

class *i* traffic are according to the following rules: (a) If $n_i^k = 0$, the next possible state n_i^{k+1} will be 0 or 1. n_i^{k+1} is 1 if a class *i* cell arrives at the (k + 1)th time slot, whose probability is λ_i . n_i^{k+1} is 0 if no class *i* cell arrives at the (k + 1)th time slot, whose probability is $1 - \lambda_i$.

(b) If $n_i^k \neq 0$, and if the class *i* cell is selected for transmission (according to rule 4), the next possible state n_i^{k+1} will be 0 or 1. n_i^{k+1} is 1 if a class *i* cell arrives at the (k + 1)th time slot, whose probability is λ_i . n_i^{k+1} is 0 if no class i cell arrives at the (k + 1)th time slot, whose probability is $1 - \lambda_i$.

(c) If $n_i^k \neq 0$, and if the class *i* cell is not selected for transmission (according to rule 4), $n_i^{k+1} = \min(n_i^k + 1, L_i)$ + 1).

(d) In each time slot, a class i cell is selected for transmission according to the priority promotion scheme described in Section 2. In other words, when the condition that $\{[(n_i \neq 0, n_i \neq L_i + 1) \text{ and } (n_g = 0 \text{ for } g < i) \text{ and } (n_k \neq L_k + 1 \text{ for } k > i)\}$ or $[(n_i = L_i + 1) \text{ and } (n_k \neq L_k + 1$ for k > i] is held, the class *i* cell is selected for transmission.

By considering all different combinations of valid changes occurred in classes 1, 2, 3, ..., N, one determines all possible next states of the current state.

Denote $p(n_i^{k+1}; n_1^k, n_2^k, ..., n_N^k)$ as the probability that the state of the class *i* cell moves from n_i^k to n_i^{k+1} given the state $(n_1^k, n_2^k, \ldots, n_N^k)$ at the kth time slot, which is defined as

$$p(n_i^{k+1}; n_1^k, n_2^k, \ldots, n_N^k)$$

$$\begin{cases} \lambda_i & \text{if } (n_i^k = 0, n_i^{k+1} = 1) \text{ or } (n_i^k \neq 0, n_i^{k+1} = 1, \\ & \text{the class } i \text{ cell is selected for service in} \\ & \text{the kth time slot)} \end{cases}$$

$$\int_{-1}^{1-\lambda_{i}} \text{ if } (n_{i}^{k} = 0, n_{i}^{k+1} = 0) \text{ or } (n_{i}^{k} \neq 0, n_{i}^{k+1} = 0, \text{ the class } i \text{ cell is selected for service in}$$

the kth time slot)
1 if
$$(n_i^k \neq 0, n_1^{k+1} = \min(n_i^k + 1, L_i + 1))$$
, the
class *i* cell is not selected for service in
the kth time slot)

(7)

The decision of which class of traffic is selected for service is based on the service discipline mentioned in Section 2. The transition probability from $(n_1^{k}, n_2^{k}, ..., n_N^{k})$ to $(n_1^{k+1}, n_2^{k+1}, ..., n_N^{k+1})$ is denoted as $q(n_1^{k+1}, n_2^{k+1}, ..., n_N^{k+1})$ is denoted as $q(n_1^{k+1}, n_2^{k+1}, ..., n_N^{k+1})$, $n_2^{k}, ..., n_N^{k}$, which is a product of $p(n_1^{k+1}; n_1^{k}, n_2^{k}, ..., n_N^{k})$, $p(n_2^{k+1}; n_1^{k}, n_2^{k}, ..., n_N^{k})$, $\dots, p(n_N^{k+1}; n_1^{k}, n_2^{k}, ..., n_N^{k})$, or

$$q(n_1^{k+1}, n_2^{k+1}, \dots, n_N^{k+1}; n_1^k, n_2^k, \dots, n_N^k) = p(n_1^{k+1}; n_1^k, n_2^k, \dots, n_N^k) * p(n_2^{k+1}; n_1^k, n_2^k, \dots, n_N^k)$$

• · · · * $p(n_N^{k+1}; n_1^k, n_2^k, \dots, n_N^k)$ (8)

7.2 Procedure

generate_state_space_and_transition_probability This procedure is used to generate all possible states and the transition probabilities. We use two lists, Open list and Closed list, to keep track of progress through the state space. Open list records the newly generated states, while Closed list records the states that have already been examined.

Procedure

Initialise: Open = [start]; Closed = [];

while Open ≠ [] do begin

remove the leftmost state from Open, call it X; generate all the next states of X according to the four rules described above, and generate the transition probability between X and its next possible states according to eqns. 7 and 8

put X on Closed; eliminate any next state of X already on either Open or Closed, as these will cause infinite loop; put the remaining next states of X, on the right end of Open;

end

From this procedure, all of the valid states and the state transition probabilities are generated.

An example illustrates the procedure. Consider an ATM network with three classes of traffic and the access time limit for each class is 3. To generate the complete state space, choose (0, 0, 0) as the initial state according to the procedure described in Section 3. The next possible states of (0, 0, 0) are (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), and (1, 1, 1), whose corresponding transition probability are $(1 - \lambda_1)(1 - \lambda_2)(1 - \lambda_3)$, $(1 - \lambda_1)\lambda_2(1 - \lambda_3)$, $(1 - \lambda_1)\lambda_2(1 - \lambda_3)$, $(1 - \lambda_1)\lambda_2(1 - \lambda_3)$, $\lambda_1(1 - \lambda_2)\lambda_3$, $\lambda_1(1 - \lambda_2)\lambda_3$, $\lambda_1(2 - \lambda_3)$, and $\lambda_1\lambda_2\lambda_3$. After the first while-loop of the procedures (0, 0, 0) is also in Closed list, it is not added to Open list. State (0, 0, 1) is chosen from Open list as the working state X to generate the next possible states, which are (0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 0, 0), (1), (1), (1, 0, 0), (1), (1), (1, 0, 0), (1), (1), (1, 0, 0), (1), (1), (1, 0, 0), (1), (1), (1), (1), 0), (1), (1), (1), 0), (1), (1), (1), 0), (1), (1), (1), 0), (1), (1), 0), (1), (1), (1), 0), (1), (1), (1), 0), (1), (1), (1), 0), (1), (1), (1), 0), (1), (1), 0), (1), (1), 0), (1), (1), 0), (1), (1), (1), 0), (1), (1), (1), 0), (1), (1), (1), 0), (0), (1),

 $-\lambda_3$, $\lambda_1(1-\lambda_2)\lambda_3$, $\lambda_1\lambda_2(1-\lambda_3)$, and $\lambda_1\lambda_2\lambda_3$. Because the next possible states of (0, 0, 1) are all in Open list or in Closed list, they are not added to Open list. Repeating the while-loop until no more state contained in Open list, the complete state space and the transition probability matrix are generated. The first four while-loop snapshots of the procedure are depicted in Table 2.

	-			• •			
Table 2:	First	tour	snai	pshots	OT.	proce	adur

Present state X	Next state	Transition probability	Open list	Close list
(0, 0, 0)	(0, 0, 0) (0, 0, 1) (0, 1, 0) (0, 1, 1) (1, 0, 0) (1, 0, 1) (1, 1, 0) (1, 1, 1)	$\begin{array}{c} (1-\lambda_1)(1-\lambda_2)(1-\lambda_3)\\ (1-\lambda_1)(1-\lambda_2)\lambda_3\\ (1-\lambda_1)\lambda_2(1-\lambda_3)\\ (1-\lambda_1)\lambda_2\lambda_3\\ \lambda_1(1-\lambda_2)(1/\lambda_3)\\ \lambda_1(1-\lambda_2)\lambda_3\\ \lambda_1\lambda_2(1-\lambda_3)\\ \lambda_1\lambda_2\lambda_3 \end{array}$	[(0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)]	[{0, 0, 0}]
(0, 0, 1)	(0, 0, 0) (0, 0, 1) (0, 1, 0) (0, 1, 1) (1, 0, 0) (1, 0, 1) (1, 1, 0) (1, 1, 1)	$\begin{array}{l} (1-\lambda_1)(1-\lambda_2)(1-\lambda_3)\\ (1-\lambda_1)(1-\lambda_2)\lambda_3\\ (1-\lambda_1)\lambda_2(1-\lambda_3)\\ (1-\lambda_1)\lambda_2\lambda_3\\ \lambda_1(1-\lambda_2)(1/\lambda_3)\\ \lambda_1(1-\lambda_2)\lambda_3\\ \lambda_1(1-\lambda_2)\lambda_3\\ \lambda_1\lambda_2\lambda_3\\ \lambda_1\lambda_2\lambda_3 \end{array}$	[(0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)]	[(0, 0, 0), (0, 0, 1)]
(0, 1, 0)	(0, 0, 0) (0, 0, 1) (0, 1, 0) (0, 1, 1) (1, 0, 0) (1, 0, 1) (1, 1, 0) (1, 1, 1)	$\begin{array}{c} (1-\lambda_1)(1-\lambda_2)(1-\lambda_3)\\ (1-\lambda_1)(1-\lambda_2)\lambda_3\\ (1-\lambda_1)\lambda_2(1-\lambda_3)\\ (1-\lambda_1)\lambda_2\lambda_3\\ \lambda_1(1-\lambda_2)\lambda_3\\ \lambda_1(1-\lambda_2)\lambda_3\\ \lambda_1(1-\lambda_2)\lambda_3\\ \lambda_1\lambda_2(1-\lambda_3)\\ \lambda_1\lambda_2\lambda_3 \end{array}$	[(0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)]	[(0, 0, 0), (0, 0, 1), (0, 1, 0)]
(0, 1, 1)	(0, 0, 2) (0, 1, 2) (1, 0, 2) (1, 1, 2)	$(1 - \lambda_1)(1 - \lambda_2) (1 - \lambda_1)\lambda_2 \lambda_1(1 - \lambda_2) \lambda_1\lambda_2$	[(1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1), (0, 0, 2), (0, 1, 2), (1, 0, 2), (1, 1, 2)]	[(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1)]

IEE PROCEEDINGS-I, Vol. 140, No. 4, AUGUST 1993